Near Wellbore Streamline Modeling for Advanced Well Completions

by

Justin Harry Skinner

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Engineering

Oil and Gas Engineering

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

May 2011

St. John's, Newfoundland and Labrador, Canada

Abstract

The following thesis presents a novel approach to streamline modeling in the near wellbore region of an oil well. Streamline modeling has existed within the petroleum field since the 1950's, with many recent advances occurring throughout the last 25 years. Today, streamline modeling is used for a variety of purposes, including flow visualization, transport calculations, reservoir simulation, history matching and many more useful applications.

In the known literature streamline modeling has not been performed within the near wellbore region of an oil well. The research presented in this thesis focuses on modeling in this vicinity, showing how near wellbore heterogeneities affect flow distribution and overall production. The research also utilizes the near wellbore streamline modeling methodology to establish a novel technique for evaluating well completion strategies. Through the use of optimal design techniques, the various parameters affecting production are analyzed and optimal completion techniques are established.

While the majority of this research focuses on a two dimensional isotropic heterogeneous media, discussion is provided for the extension to three dimensional modeling. In addition, discussion is also provided for the general case of an anisotropic heterogeneous reservoir medium.

Various topics for additional work as a result of this research are provided, including new particle tracking techniques, connection factor implementation and the extension to more complex physics.

Acknowledgements

The completion of this thesis would not have been possible without the help of numerous individuals. First and foremost, I would like to thank my supervisor, Dr. Thormod Johansen. His guidance, knowledge and full support has allowed me to excel in my chosen field and has greatly enhanced my capabilities as an engineer. This research would not have been possible without his unwavering guidance, and for that I am grateful. I would also like to acknowledge my co-supervisor, Dr. Faisal Khan, for his guidance and financial support.

The constant love and support of my family must also be recognized. My parents have supported me throughout my life, guiding me to the right choices and supporting each decision I have made. My parents have taught me more about life than can ever be learnt in a degree, and have molded me into the person I am today. I am forever grateful for your support, and I hope my work and accomplishments have made you proud. I am also grateful for my two sisters, Sonya and Tara, and my two nieces, Hayley and Adria, who have brought me a wealth of cheer and happiness throughout the years. The support of my family has enriched my life, and for that I am thankful.

Several other professors at Memorial University must be recognized for their continuous help and support. Dr. Leonard Lye and Dr. Stephen Bruneau have provided me with guidance concerning academics, career choices and life in general. They have been my references on numerous scholarships and applications, always willing to sacrifice their time to lend a helping hand. I truly appreciate all of their help over the years. I have enjoyed every discussion and am grateful for the advice and opportunities they have provided.

I must also thank my peer and friend Sameena Trina. Sameena has helped me with numerous technical issues throughout my research, but her most valuable asset has been her friendship. It has been a pleasure working with Sameena, and I will cherish the times spent working and laughing together.

In addition, I must thank the numerous organizations that have contributed to my financial support over the past two years. These include the Natural Sciences and Engineering Research Council of Canada (NSERC) for awarding me the Alexander Graham Bell Canadian Graduate Scholarship (CGS-M), the Research and Development Corporation of Newfoundland and Labrador (RDC) for awarding me the Ocean Industries Student Research Award, the Department of Education on behalf of the Government of Newfoundland and Labrador for awarding me the Graduate Scholarship in Resource Development, and the School of Graduate Studies on behalf of Memorial University for awarding me the Dean's Excellence Award and the George Hatcher Memorial Scholarship. I would also like to thank the graduate office at the Faculty of Engineering and Applied Science for providing me with numerous graduate teaching assistantships which have supplemented my income as well as provided me with valuable teaching experience.

Last but not least, I would like to thank Moya Crocker for her assistance throughout my program. Her help has been invaluable in progressing with the formalities of my degree, and her friendliness throughout the process is greatly appreciated.

Table of Contents

List of Tables

List of Figures

Nomenclature

A	=	area
$\vec{A}$	=	Laplace coefficient matrix
a	=	node distance constant
a_r	=	radial velocity coefficient
a_z	=	vertical velocity coefficient
a_θ	=	tangential velocity coefficient
$a(i,j)$	=	Laplace coefficient belonging to pressure node $p_{i,j}$
$\vec{B}$	=	Laplace boundary condition matrix
b_r	=	radial velocity constant
b_z	=	vertical velocity constant
b_θ	=	tangential velocity constant
$b(i,j)$	=	Laplace coefficient belonging to pressure node $p_{i,j+1}$
$c(i,j)$	=	Laplace coefficient belonging to pressure node $p_{i,j-1}$
$d(i,j)$	=	Laplace coefficient belonging to pressure node $p_{i-1,j}$
D	=	hydraulic diameter
$e(i,j)$	=	Laplace coefficient belonging to pressure node $p_{i+1,j}$
$\vec{e}_r$	=	unit vector in the radial direction
$\vec{e}_z$	=	unit vector in the vertical direction
$\vec{e}_\theta$	=	unit vector in the tangential direction
h	=	reservoir thickness
i	=	radial block reference / variable counter
j	=	tangential block reference / variable counter
J	=	streamlines per block
k	=	number of factors
K	=	bulk permeability
$\overline{K}$	=	permeability matrix
K_b	=	block permeability
K_c	=	crushed zone permeability
K_{ca}	=	casing permeability
K_d	=	damage zone permeability
K_{fs}	=	free space permeability

K_n	=	generalized directional permeability
K_p	=	perforation permeability
K_r	=	permeability in radial direction
K_t	=	permeability in tangential direction
K_x	=	permeability in x direction
K_y	=	permeability in y direction
K_z	=	permeability in z direction
K_θ	=	mathematical value relating K_r and K_t
L	=	length
L_{in}	=	entrance length between streamlines along a radial grid block boundary
L_{out}	=	exit length between streamlines along a radial grid block boundary
L_p	=	perforation length
M	=	number of tangential blocks
n	=	number of block generators / degree of fraction
N	=	number of radial blocks
n_x	=	unit vector in x direction
n_y	=	unit vector in y direction
n_z	=	unit vector in z direction
p	=	pressure
p_e	=	external reservoir pressure
p_{if}	=	internal face pressure
p_o	=	open-hole skin free pressure
p_{wf}	=	wellbore pressure (bottom-hole pressure)
$\vec{P}$	=	Laplace pressure matrix
q	=	flow rate
$q_{i+1/2}$	=	flow rate between adjacent radial blocks
Q	=	total flow rate
r	=	radius (polar position variable)
r_d	=	damage zone radius
r_e	=	external reservoir radius
r_{en}	=	streamline block entrance radius
r_{ex}	=	streamline block exit radius

r_i	=	grid node radius
$r_{i\pm1/2}$	=	grid boundary radius
r_{if}	=	internal face radius
r_w	=	wellbore radius
Re	=	Reynolds number
S	=	total skin
S_d	=	formation damage skin
S_l	=	splitting line length
S_{pc}	=	pseudoskin factor
S_{pd}	=	convergence ski
S_{pl}	=	perforation length skin
S_{pp}	=	partial penetration skin
t	=	time
t_{D_i}	=	dimensionless time
t_{min}	=	minimum time or time of flight
t_p	=	perforation thickness
t_r	=	radial travel time
t_z	=	vertical travel time
t_θ	=	tangential travel time
T	=	total time of flight along a streamline
$\vec{u}$	=	volumetric flux vector or Darcy velocity
u_r	=	volumetric flux in radial direction
u_z	=	volumetric flux in vertical direction
u_θ	=	volumetric flux in tangential direction
V	=	velocity
$\bar{V}_p$	=	arbitrary pore volume
x	=	Cartesian position variable
x_{D_i}	=	dimensionless length
y	=	Cartesian position variable
z	=	Cartesian position variable
z_{en}	=	streamline block entrance plane
z_{ex}	=	streamline block exit plane

α	=	significance level
β	=	formation volume factor
ξ	=	streamline segment length
θ	=	angle (polar position variable)
θ_{en}	=	streamline block entrance angle
θ_{ex}	=	streamline block exit angle
θ_x	=	angle with respect to x axis
θ_y	=	angle with respect to y axis
θ_z	=	angle with respect to z axis
θ^*	=	interpolated angle with a tangential velocity equal to zero
λ	=	mobility
λ^r	=	upscaled mobility in radial direction
λ^t	=	upscaled mobility in tangential direction
μ	=	oil viscosity
ρ	=	oil density
τ	=	time of flight
$\emptyset$	=	porosity
∇	=	gradient of a vector in Cartesian coordinates
∇_{r}	=	gradient in radial direction
∇_{θ}	=	gradient in tangential direction
$\nabla_{\mathrm{r},\theta}$	=	gradient of a vector in polar coordinates
ΔP_{skin}	=	pressure loss attributed to skin

- Chapter 1 -

1.0 Introduction to Reservoir and Streamline Modeling

This chapter is meant to serve as an introduction to the material to be followed throughout this thesis. The chapter starts with the large scope of the petroleum engineering field and slowly narrows the focus to the area of detailed near wellbore streamline modeling; the subject of the thesis. The roles of a reservoir engineer are discussed, followed by an explanation of reservoir modeling and streamline modeling. The extent of the modeling procedures are also discussed, looking at the differences between near wellbore models and full field models. In addition, the current methodology for dealing with wells within streamline models is explained to provide greater context for the novelty of the work to be presented. The purpose of this research is discussed towards the end of the chapter, followed by an explanation of the thesis layout to follow.

1.1 The Field of Petroleum Reservoir Engineering

The field of petroleum engineering deals with all activities concerning the production of oil and natural gas. The field is often divided into the upstream and downstream sectors. Upstream operations are concerned with exploration and production, while downstream operations deal with refining, upgrading and distribution. A more detailed breakdown of the industry into three components is provided in Figure 1. The focus of this research lies within the upstream sector, dealing with specific aspects of petroleum production.

Figure 1 - Petroleum Industry Breakdown

Within the upstream production segment numerous engineering roles are required. Common roles include drilling engineering, facility engineering, subsea engineering, production engineering and reservoir engineering. Each performs specific roles related to exploration, production and development. The work to be presented in this thesis is suited for both production and reservoir engineers, as their fields are highly related.

The work of a reservoir engineer focuses on understanding a reservoir usually located kilometers beneath the surface of the earth; be it on land or at sea. The primary roles of a reservoir engineer include the estimation of oil-in-place, determining the potential recovery and attaching a time frame to that recovery. Reservoir engineers are also responsible for maximizing recovery; a task which requires an in-depth understanding of

the reservoir, its piping and its various limitations. A reservoir engineer must utilize an understanding of geology, applied mathematics and the laws of physics and chemistry to assess the volume of oil held within a reservoir and its potential for flow. An understanding of fluid mechanics and transport phenomena within a reservoir are crucial to the development and success of a field.

Working hand in hand with reservoir engineers are production and surveillance engineers. These engineers focus on the daily optimization of production and injection rates, looking to manage and enhance the production of a field. A significant portion of their work is directly related to the reservoir, especially in areas concerning production optimization, completion strategies and the like. This thesis deals with modeling of flow throughout a reservoir and into well completions. Therefore it is ideally suited for individuals with backgrounds or interest in production and reservoir engineering.

1.2 Reservoir Modeling

The most crucial task of a reservoir engineer is the development of a reservoir model. This is the tool that allows him/her the ability to understand a reservoir and perform key tasks concerning volumes, recovery and optimization. A reservoir model is a complex computer program utilizing applied mathematics to create a simulation of a reservoir. These models attempt to incorporate all physical and chemical characteristics related to fluid volumes and production. The models integrate all available data to determine the most likely scenario regarding what can be expected from the reservoir, with the production profile being the essential output. The reservoir model attempts to simulate the behavior of fluids within the reservoir, and through careful alteration of variables determine an optimal strategy for future field development. Based upon the outcome of the optimized model, multiple realizations concerning the statistical likelihood of occurrence are generated. Economic decisions concerning the future of a field are then based upon the reservoir simulation outputs.

Reservoir models must incorporate as much information as possible concerning geology, petrophysics, fluid behavior and the like. As well, all elements of the facility, piping and elements of the model itself must be incorporated and understood to generate reliable simulation results. Typical inputs required for a reservoir simulation model are provided in Figure 2.

Various forms of reservoir modeling may be utilized for a particular well or field. The most commonly used technique is a finite difference model. In these models a reservoir is broken up into a finite difference grid. The process generates grid blocks which are then populated with various data concerning the rock and fluid properties within each block.

In breaking the reservoir into finite grid blocks, complicated differential and partial differential equations governing flow may be solved using finite difference equations as approximations to the differentials. Sophisticated finite difference reservoir modeling programs are commercially available in the petroleum industry, with the Eclipse™ software developed by Schlumberger being the most utilized. This software has the capability of modeling various complex physics and fluid phenomena on a full field basis, making it an essential tool for production facilities worldwide. An illustration of a finite difference reservoir created within MATLAB™ by SINTEF is provided in Figure 3. The figure illustrates the permeability distribution throughout the reservoir.

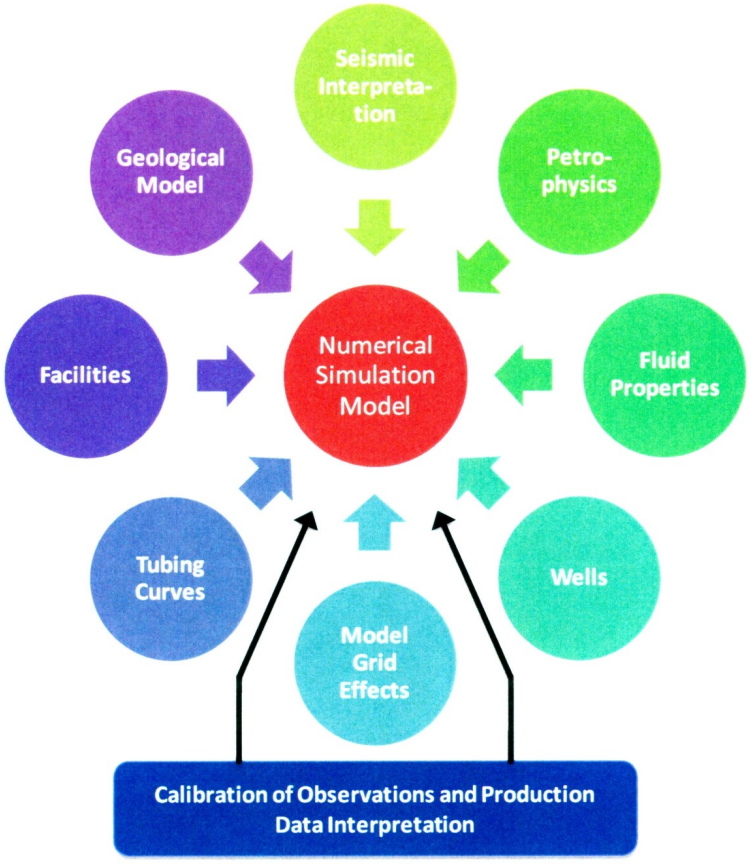


Figure 2 - General Inputs into Reservoir Models

Source: Modified from (Johansen, 2008)

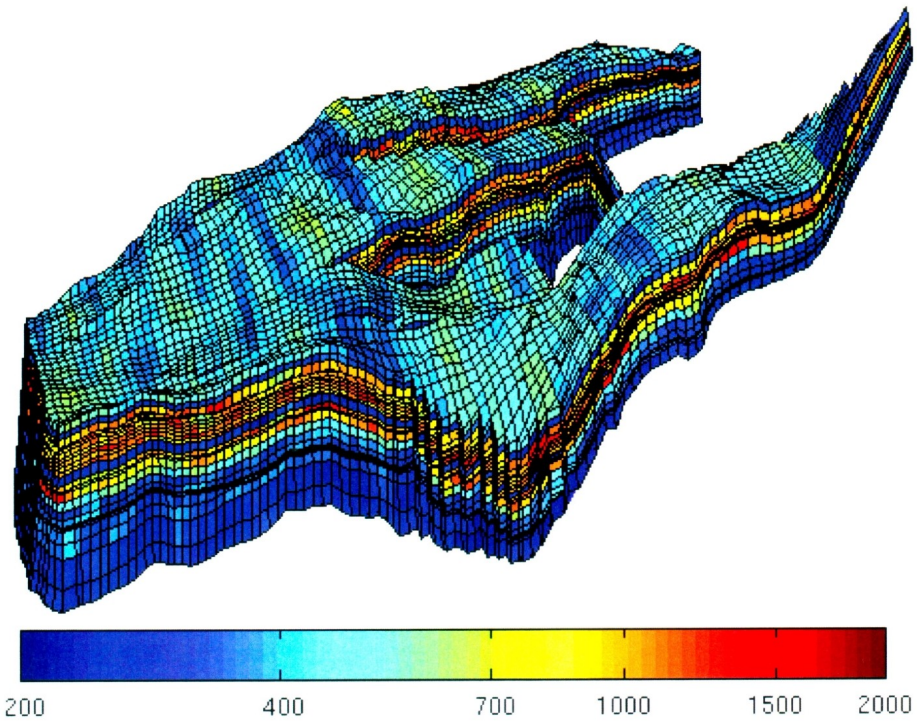


Figure 3 - Finite Difference Reservoir Model Displaying Permeability Variation

Source: (SINTEF, 2010)

While finite difference models are the standard throughout the petroleum industry, they are not the only reservoir modeling procedure available. Streamline models are also important and gaining prominence within the industry, as discussed in the next section.

1.3 Streamline Modeling

Streamlines models are prominent in numerous fields of physics involving the flow of fluids through or around objects. As such, they are commonly used in the areas of fluid dynamics and aerodynamics. Within the petroleum field, streamlines are used to aid in flow visualization, solving complex fluid transport phenomena and numerous other developing areas.

A streamline is "a line that is tangent to the instantaneous velocity vector of flow" (Thiele, 2001). In essence, they are lines mapping the velocity field throughout a fluid at a given instant in time. Terms often associated with streamlines include pathlines and streaklines. A pathline is a line that traces the path of a particle with time. In steady state flows pathlines and streamlines are equivalent. However, in unsteady flows streamlines will vary with time. This temporal variation will affect the movement of particles, forcing differences between a single pathline and any set of streamlines. Streaklines are lines created by connecting all particles that travel through the same point at an earlier instant in time. As with pathlines, for steady state flow streamlines and streaklines are equivalent. Streamlines, pathlines and streaklines are all different techniques used to visualize flow patterns, and providing flow is steady, they will all be the same.

An important principle in streamline modeling is the fact that streamlines can never cross. Streamlines represent the velocity field at a given instant in time. Velocity is a vector composed of a both magnitude and direction. At any one point in space only one velocity vector can exist, meaning flow can only go in one direction. As a result, streamlines cannot cross. Figure 4 provides a simple illustration of streamlines moving from left to right around a cylindrical object.

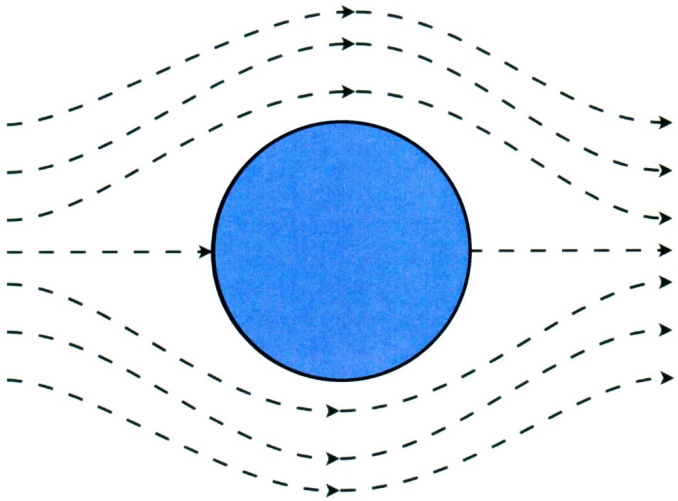


Figure 4 - Streamlines around a Cylinder

Source: Modified from (National Aeronautics and Space Administration, 2010)

Within the petroleum industry streamline modeling has been utilized since the 1950's. A history of the developments within the field is provided in Chapter 2. Today, streamline modeling is being used for numerous purposes within the petroleum field. Modern models provide a means of efficient flow visualization that may be critical in understanding flow patterns and stream distributions within a reservoir. In addition, streamlines are utilized in solving fluid transport problems involving front propagation, especially when reservoir heterogeneity has a significant impact on flow. Streamlines are now being used for various other purposes within the reservoir field, including dynamic reservoir characterization, upscaling, rate allocation, history matching and the like (Datta-Gupta & King, 2007). These uses are all possible because of two simple principles of streamline modeling; its speed and its accuracy. An illustration of a full field streamline modeling output showing flow profiles between various wells is provided in Figure 5.

Figure 5 - 2D and 3D Perspective of Full Field Streamline Models

Source: (Thiele et al., 2007)

The streamline modeling technique is best suited for reservoirs in which flow profiles are dominated by reservoir heterogeneity in convection dominated flows. Once streamlines are generated for an entire reservoir they define the spatial variation in the fluid flow. Streamlines will tend to concentrate in areas of high permeability, with faster flow in these regions. Likewise, streamlines will have less resolution for areas of low permeability, with slower flow. The resultant streamline distribution will reflect the permeability distribution throughout the reservoir.

An important concept in streamline modeling is "time of flight". Time of flight refers to the time it takes a chemically neutral particle to travel along a streamline. The time of flight concept, first developed by Pollock (1988), has greatly affected the development of streamline models within the petroleum industry. In its simplest form, areas of high permeability will have small times of flight, meaning faster flow. Likewise, areas of low permeability will have large times of flight, meaning slower flow. This is significant in

understanding the movement of flood fronts during water flooding and understanding the effects of channeling within a reservoir.

Another significant development in streamline modeling within the petroleum field is the approximation of solving three dimensional fluid flow in a reservoir based upon a single dimensional calculation along streamlines. This allows streamlines and subsequent calculations to be evaluated independent of the underlying finite difference grid and corresponding reservoir heterogeneities. It is this separation that is responsible for the significant speed improvements as compared to the traditional finite difference models. The validity of the underlying assumptions involved within this technique with respect to near wellbore streamline modeling is discussed in Chapter 5.

1.4 Streamline Modeling versus Finite Difference Modeling

Streamline models and finite difference models cover similar areas in their outputs, but they are not interchangeable options. Several significant differences exist between the two modeling techniques, with each having their own benefits and downfalls.

As discussed, a finite difference model breaks a reservoir up into a finite difference grid. Grid blocks are populated with fluid and reservoir properties and equations are used to calculate the flux of fluids from block to block. The end result provides excellent data with respect to volume calculations and fluid locations, but it does not provide insight into the precise movement of the fluids.

Streamline simulation provides precise information with respect to the movement of fluids, showing the direction and speed of particles throughout a reservoir. The convective physics of flow are modeled providing additional insight into the reservoirs behavior. Therefore not only are the volumes and fluid locations known, but the direction of the fluids movement is also known. It may be assessed that in comparing streamline modeling with finite difference modeling for convection dominated reservoirs, nothing is lost but more is gained (Datta-Gupta & King, 2007).

While streamline models do provide additional insight into reservoir behavior, several disadvantages do exist. The first issue lies in the modern approach to streamline modeling of fluid transport problems. In these problems saturation calculations are transformed from three dimensional space variables to one dimensional problems along streamlines, with time of flight being the single dependent variable. The transformation from three dimensional problems to single dimensional problems has several underlying

assumptions. As a result, applying a transform from three dimensions to time of flight and back again will lead to mass conservation issues. The alternative solution for volume calculations involves using a stream tube approach by tying together adjacent streamlines. This can be numerical exhaustive and complex for three dimensional problems. This issue is elaborated upon further in Chapter 5.

Another significant downfall of the streamline modeling approach involves time discretization. If flow in a reservoir is not steady state then streamlines must be constantly updated. While research into time stepping is taking place, an accurate solution to this problem concerning error estimation and convergence between streamlines still requires further investigation (Datta-Gupta & King, 2007).

Finally, the most significant downfall of streamline modeling when compared to finite difference modeling lies in the scope of its physics. Streamline models are best suited for reservoirs in which flow is convection dominated. Other physical processes involving gravity, capillary pressure, miscibility, phase behavior and the like are currently not well suited for streamline modeling. Research has been and is currently being done within these areas, but it has not progressed to the state of being well suited for incorporating all physical and chemical mechanisms which may occur. Finite difference programs are available to model complex physics within reservoirs and are currently much better suited for these scenarios. Figure 6 provides a graph showing the applicability of the two different modeling techniques.

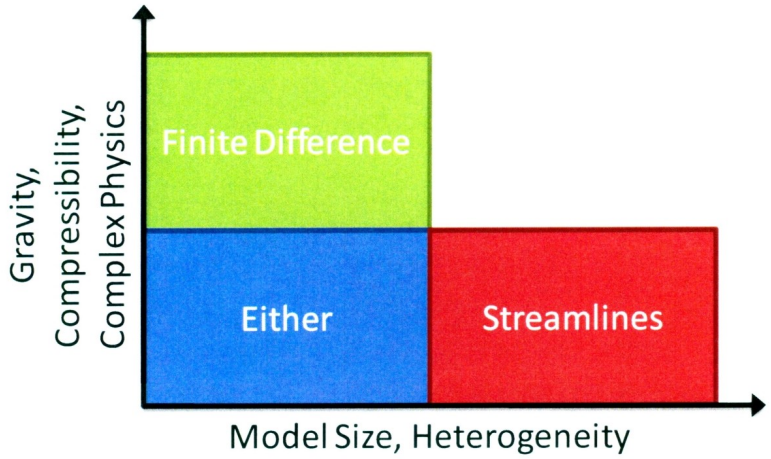


Figure 6 - Applicability of Streamline Modeling vs. Finite Difference Modeling

Source: Modified from (Datta-Gupta & King, 2007)

Streamline models do hold a distinct advantage over finite difference models in terms of run times. Finite difference programs are much slower, involving many more calculations from grid block to grid block. Streamline models are much simpler and quicker, simplifying to one dimensional equations along streamlines for various problems. Therefore, when the number of grid blocks is high and heterogeneity is large, streamlines may be a more viable option. This is especially evident for reservoir characterization and upscaling, in which the growth in geological modeling is far exceeding the growth in finite difference capabilities.

In summary, streamline models provide additional insight into the nature of flow within a reservoir. They are best suited for convection driven reservoirs. They cannot replace finite difference simulation due to an inability to reliably model complex physical and chemical mechanisms. As such, both modeling techniques are best used together, providing a reservoir engineer with maximum information to enhance decision making capabilities.

1.5 Full Field Reservoir Modeling versus Near Wellbore Modeling

Reservoir modeling is traditionally broken up into two distinct scales; near wellbore modeling and full field modeling. The choice between these two scales depends on the life-stage of a field and/or well and the intended output of the simulation process. Figure 7 is a plot illustrating the difference in scale between these two modeling techniques. As illustrated, the full extent of a near wellbore model may fit within a single grid block of a traditional full field model.

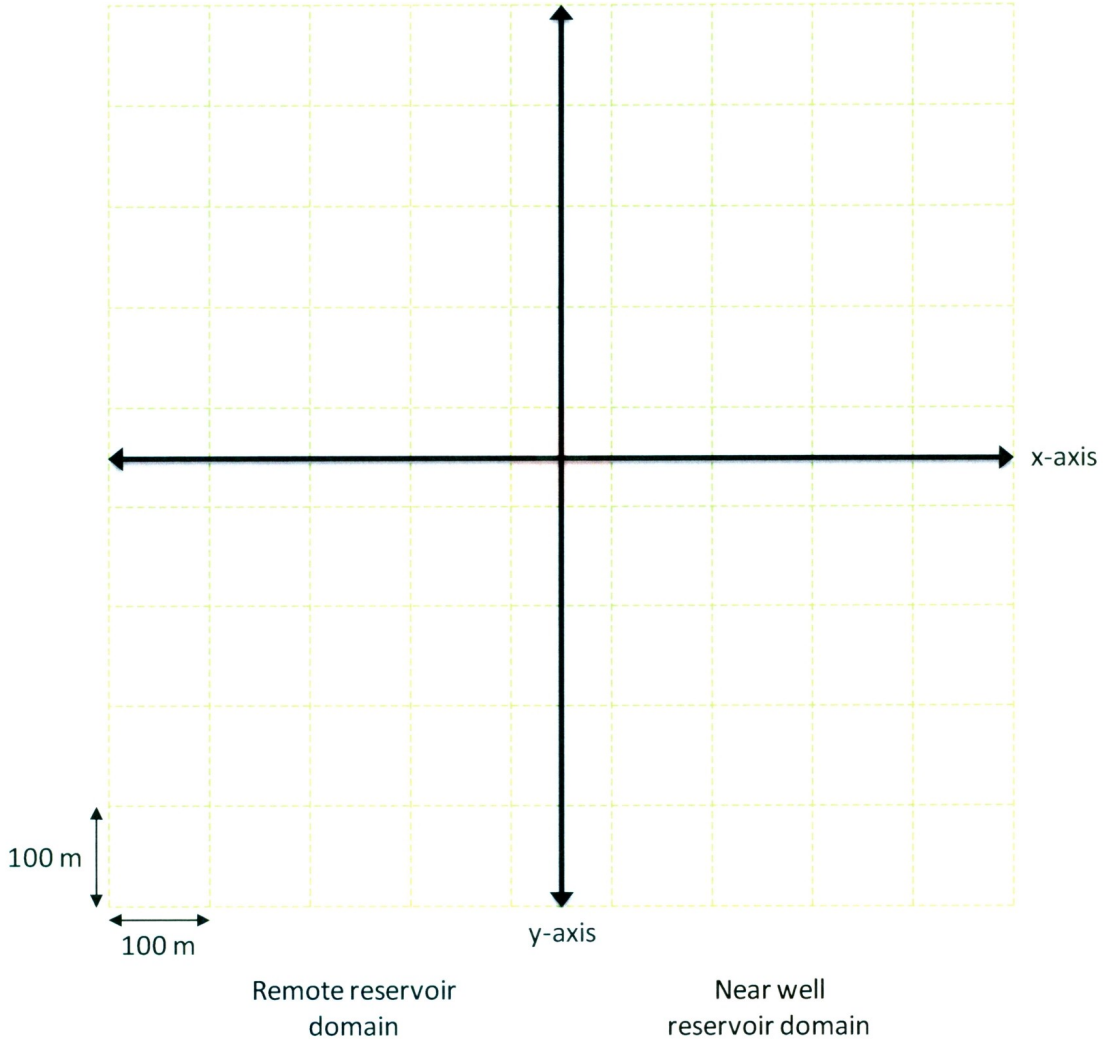


Figure 7 - Full Field Scale versus Near Wellbore Scale

Near wellbore models are most often utilized early in the development of a field. They are best suited to provide information concerning a well's potential flow when the well is first being planned, on early production or unaffected by surrounding wells. Near wellbore models are based on much higher density grids in a limited region surrounding a well. They provide valuable information concerning the magnitude of near wellbore effects and their potential influence on production. Near wellbore models may be used to evaluate the best completion strategy to provide optimal production. This is an intended goal of this research.

As a field matures and more wells are planned and implemented, the importance of near wellbore models is reduced. Full field models covering a much larger domain with multiple wells then gain in prominence, with their output being critical to the economic success of a field. Figure 8 provides a plot outlining the success of the modeling techniques over the life of a field.

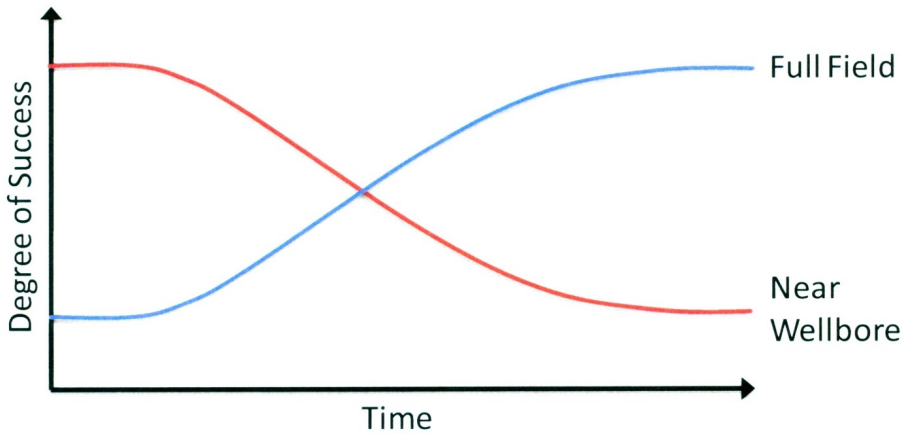


Figure 8 - Success of Near Wellbore and Full Field Modeling Techniques

Source: Modified from (Johansen, 2008)

1.6 Treatment of Wells in Streamline Models

In the current streamline models utilized in the petroleum industry there are three ways to simulate wells. Before they can be discussed, the nature of wells within these models must be explained. To represent a well, a sink or source term (injector or producer) is added within a Cartesian grid block. As explained previously, for full field modeling these blocks are large in comparison to the size of the well. Through the streamline modeling process the velocity across all grid blocks are known, and linear interpolation is used to determine the velocity within the grid blocks. However, when a grid block contains a well, this creates significant errors. As can be seen in Figure 9, linear interpolation for a cell containing a well may lead to significant errors as a result of sign differences and the nonlinear increase in velocity as fluids approach or flow away from a well.

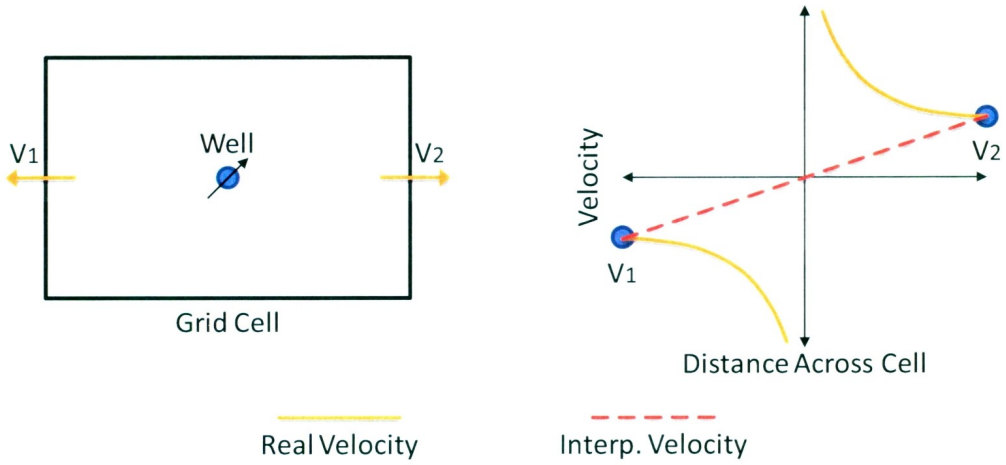


Figure 9 - Interpolation Error in Cells with Wells

Source: Modified from (Datta-Gupta & King, 2007)

Three methods are utilized within the literature to counteract this issue and model streamlines in cells containing wells. The first is to simply terminate streamlines once the

boundary of a grid block containing a well is reached. This is by far the most commonly used method and involves neglecting the pore volume attributed with grid blocks containing wells. The procedure results in errors of less than 10% with respect to time of flights along streamlines (Datta-Gupta & King, 2007).

A second approach is proposed by Liu et al. (1999). In this method the velocity within a cell is calculated based upon a weighted average assuming the grid block containing the well to be in an infinite homogenous reservoir. The velocity is based upon the known face velocities and the known well velocity. The approach can be modified to handle wells located on the face of a cell as oppose to within a cell (Datta-Gupta & King, 2007).

A third approach proposed by Datta-Gupta and King (2007) involves local refinement within cells with wells. The refinement allows for an analytical treatment to provide reasonable interpolation results between cell boundaries and wells. The method divides a cell containing a well into four quadrilaterals with the point of intersection representing the well. The well is then treated as a line source singularity in which flux is maintained into the well but area vanishes. Total influx is the sum of the influx from all four quadrilaterals. While the method is not fully rigorous, it satisfies the boundary conditions and shows how velocities become singular near wells (Datta-Gupta & King, 2007).

In each of these methods rough approximations are used to model flow into a well. The grid block is considered homogenous and near wellbore effects cannot be modeled. To truly understand the nature of flow near the well, significant grid refinement must be completed within the vicinity, highlighting the need for near wellbore models.

1.7 Near Wellbore Damage

Within the near wellbore region various factors may influence the production of a well. These factors include petrophysical properties, fluid properties, extent of formation damage, stimulation treatments, well geometry, well completions, number of fluid phases and flow velocity types (Yildiz, 2006). The cumulative effect of these parameters on the overall flow of a well is known as total skin. In general, any effect influencing the production of a well may be conveniently modeled as skin. Skin is simply a correction factor to account for flow difference between actual flow and the flow expected for the general case of an open-hole undamaged near wellbore region. Positive skin represents a reduction in flow, while negative skin represents an increase in flow. Figure 10 shows the effect of positive skin on wellbore pressure for a constant flow rate.

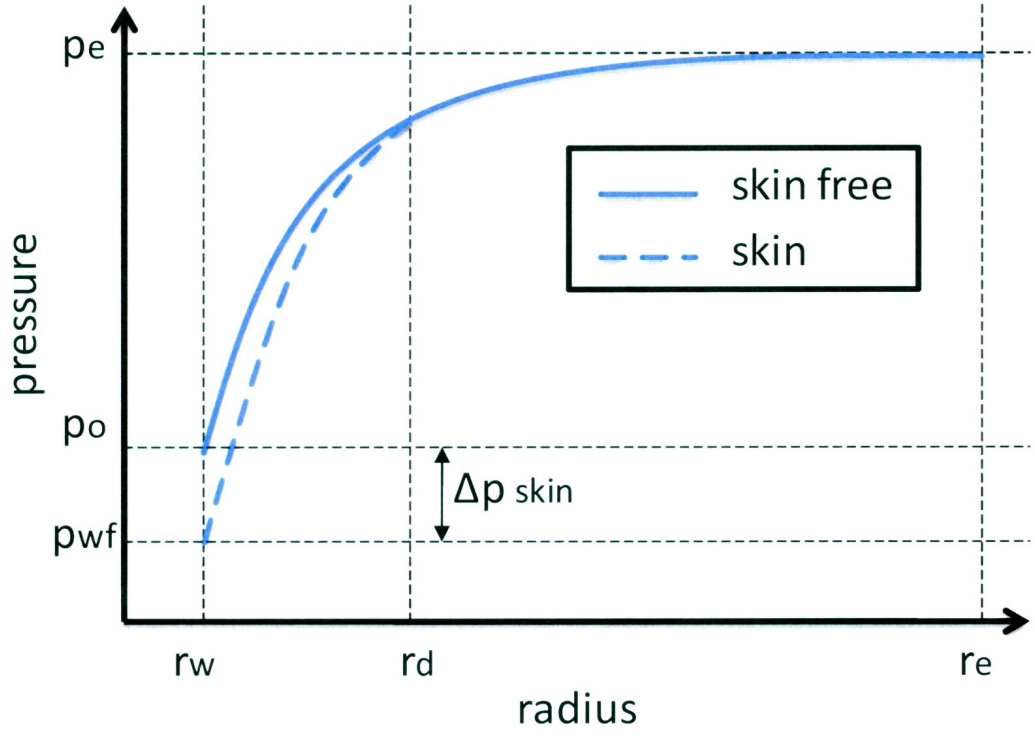


Figure 10 - Effect of Positive Skin on Wellbore Pressure

In Figure 10 r_w, r_d and r_e stand for the wellbore radius, damage radius and reservoir radius; p_{wf}, p_o and p_e stand for the wellbore pressure, open-hole skin free pressure and external boundary (reservoir) pressure; and ΔP_{skin} stands for the pressure loss attributed to skin.

Each element that contributes to flow and pressure changes may be separately analyzed, with the summation of the effects equal to the total skin. Some of the most common elements contributing to skin are discussed in this section.

The process of overbalanced drilling into a formation leads to an influx of fluids into the reservoir. These fluids include drilling mud, cement, solids and completion fluids. The net effect is a reduction in permeability surrounding a well. This is known as formation damaged and is displayed in Figure 11. Formation damage causes an increase in pressure loss, leading to a positive skin component.

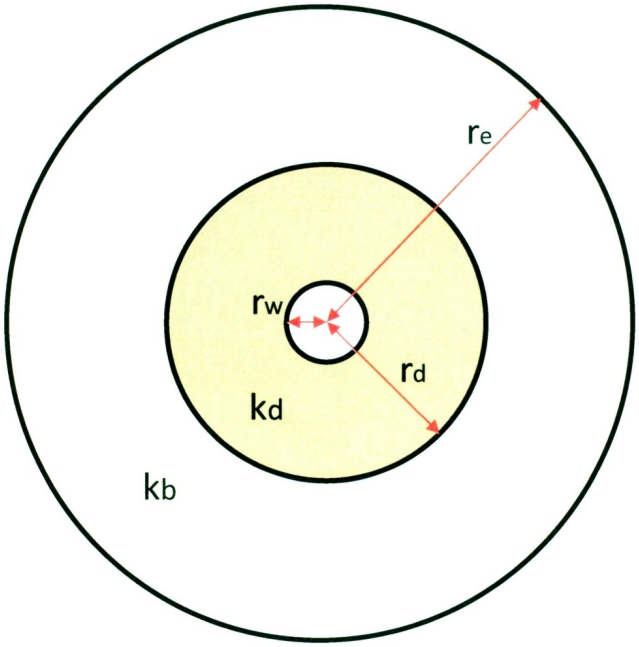


Figure 11 - Skin as a Result of Formation Damage

The terms in Figure 11 refer to damage radius (r_d), damage zone permeability (k_d) and bulk formation permeability (k_b). Skin as a result of formation damage is calculated via the Hawkin's formula, as follows:

$$S_d = \left(\frac{k_b}{k_d} - 1\right) ln\left(\frac{r_d}{r_w}\right)$$

In an open-hole completion for a fully completed reservoir, flow is radial. Completion activities that cause flow to veer from radial result in additional pressure loss, thereby further increasing the total skin. In many well completions cemented casing and perforations are utilized. In the process of completing perforations high charged shots are fired into the reservoir. The perforations are meant to provide clear flow channels into the well, but the act itself will create a crushed zone around the perforation of reduced permeability. This is depicted in Figure 12. The perforation may lead to an increase or decrease in flow depending upon the overall significance of each portion. The pressure loss or gain from completion activities is known as a pseudoskin factor (S_{pc}). As shown in Figure 12, this may be further divided into skin from the perforation itself and skin from the crushed zone region surrounding it.

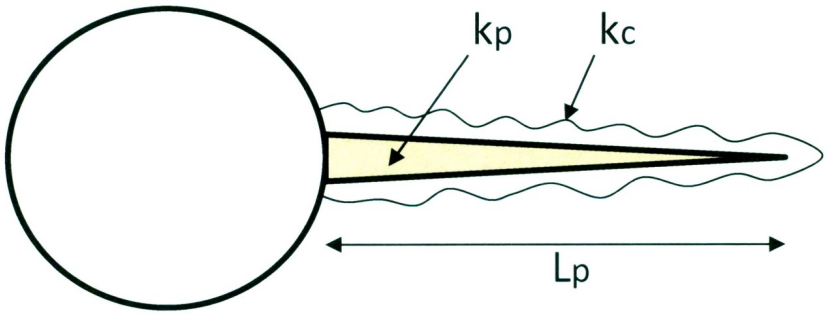


Figure 12 - Skin as a Result of Well Completions

Often the type of completion used for a well, along with heterogeneities within the near wellbore region, will result in flow patterns veering from radial. Streamlines are useful in showing these changes in flow patterns, further highlighting their significance for near wellbore modeling. For the path of a particle to turn, additional pressure loss must occur. The reduction in flow as a result of this pressure loss is known as convergence skin (S_{pd}) and is depicted in Figure 13.

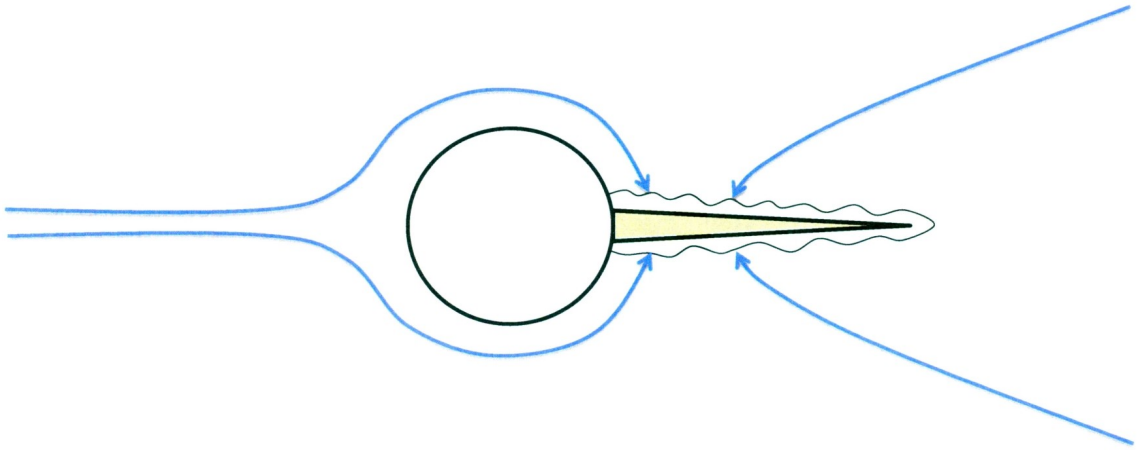


Figure 13 - Skin as a Result of Flow Convergence

An ineffective completion may also lead to additional skin. For instance, if the length of a perforation does not exceed the length of the damage zone, fluids must lose additional pressure to enter into the perforation. This is known as skin due to perforation length (S_{pl}) and is depicted in Figure 14.

In Figure 14 the terminology refers to the damage zone radius (r_d) and the length of the perforation (L_p). If L_p is less than $(r_d - r_w)$ then positive skin due to perforation length will occur. If L_p is greater than $(r_d - r_w)$ the skin may be negative (stimulation).

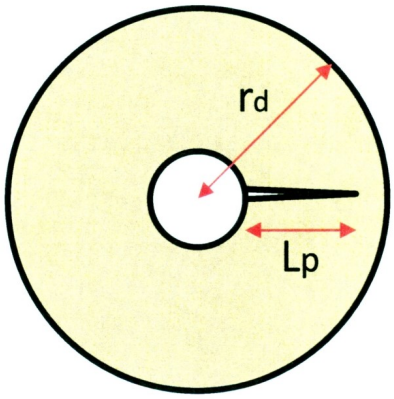


Figure 14 - Skin as a Result of Perforation Length

While skin as a result of convergence around perforations has been discussed, another form of convergence skin may occur as a result of the reservoir layer being partially completed. This is known as skin due to partial penetration (S_{pp}). The difference between the two flow convergence skins is depicted in Figure 15.

Several other sources of skin exist related to well inclination, phase behavior and stimulation techniques, but these are beyond the scope of this research. As discussed, the sum of all skin components contributes to the total skin and hence the total change in flow. The goal must be to minimize skin such that optimal flow performance is achieved. It is important to note that total skin is usually known from a well test. By understanding the components which contribute to total skin, its potential effects may be minimized.

In full field models, be they finite difference or streamline based, the effect of near wellbore damage cannot be modeled. Their inclusion can only be treated as a total skin value for a well by comparing the results expected from skin free production to the actual production rate from a well, as determined via a well test. To understand and model the nature of the components contributing to skin, and potentially minimize their influence,

near wellbore models should be invoked. Throughout this thesis near wellbore streamline models are utilized to understand the significance of near wellbore effects and determine optimal completion strategies as a result of the modeling procedure. As a result of the literature review performed it is concluded that near wellbore streamline modeling is a novel technique, not found within the published literature.

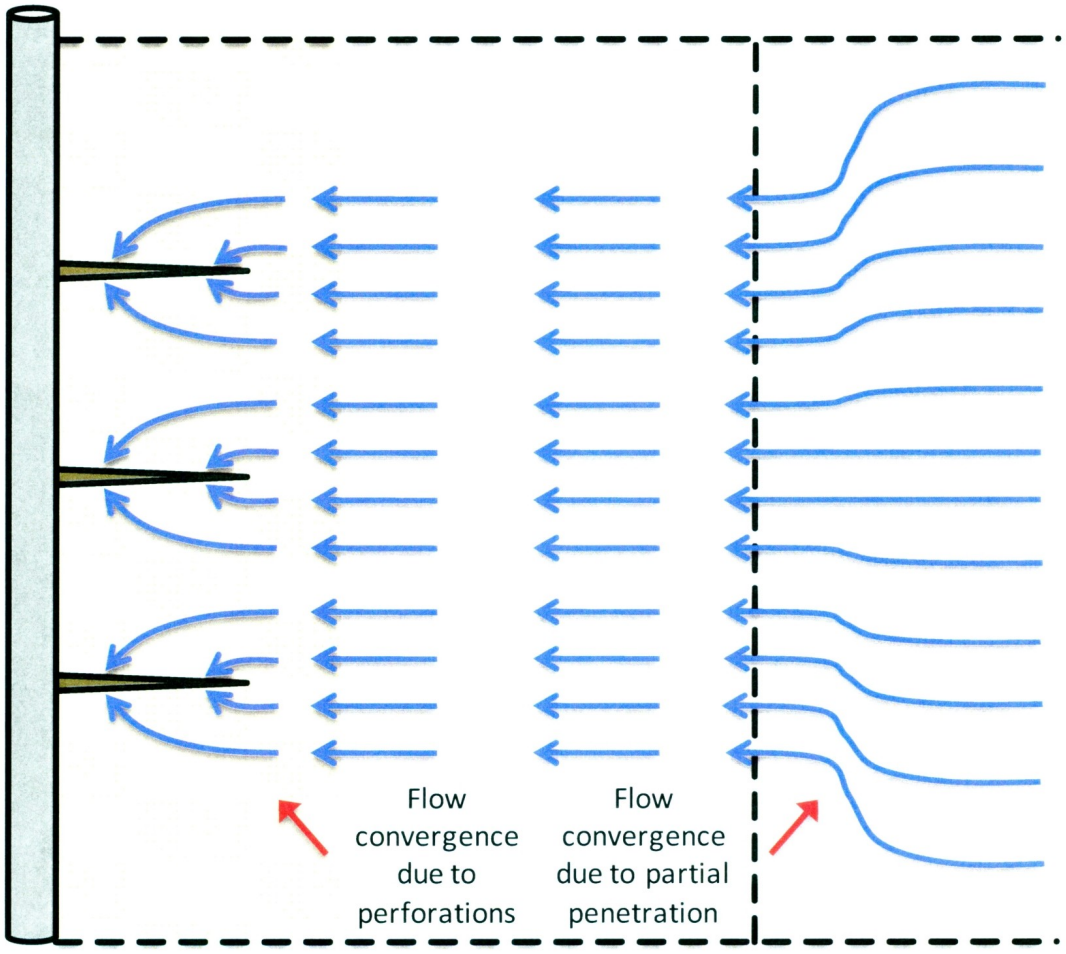


Figure 15 - Skin as a Result of Partial Penetration and Perforations

Source: Modified from (Yildiz, 2006)

1.8 Purpose of Research

Throughout this thesis various aspects of near wellbore modeling are discussed, each with its own separate purpose or reasoning. The main goal of this research is the implementation of fully functional streamline models for an isotropic heterogeneous medium on a near wellbore scale. Streamline research on a near wellbore scale is non-existent within the published literature. Therefore, this research intends to prove the validity of the approach and its potential usefulness within the petroleum industry.

A specific goal, and perhaps the most crucial for petroleum installations, is to show how near wellbore streamline models may be utilized to evaluate the effectiveness of various completion strategies. In doing so, completion design may be optimized using the streamline modeling techniques, showing how optimized parameters will influence potential flow. To illustrate these concepts three main case studies are investigated. The first two study a complex completion consisting of casing, near wellbore damage and perforations. The third studies an open-hole completion, evaluating the effect of filter cake removal on overall production.

Another goal of this research is to show how design of experiments methodology can be used to understand and simplify complex nonlinear models, as well as aid in understanding the results of detailed computer experiments. Experiment design methodology is utilized in determining the effects of various parameters within the case studies examined in this thesis, as well as determining the optimal parameters for flow optimization.

In this thesis, detailed modeling is performed for two dimensional models assuming parallel flow in the vertical direction. The extension of the modeling procedure to three dimensions is discussed and is conceptually simple, but it is more complex to implement from a programming perspective. The methodology for establishing a three dimensional model is the final goal of this research.

1.9 Thesis Structure

This thesis is divided into six distinct chapters. The current chapter provides general information pertaining to the field of reservoir engineering and the purpose of various reservoir modeling techniques. The goal is to introduce the reader to the streamline modeling process and the purpose of the research to follow in the remaining chapters.

Chapter two provides a literature review of the significant developments in the area of streamline modeling. The origins of the study of flow through porous media are discussed, followed by an explanation of techniques used prior to the modern approach. The development of the modern approach is then discussed, followed by a summary of recent research within the field.

Chapter three outlines the streamline modeling process in detail. The process methodology is discussed, focusing on a heterogeneous isotropic reservoir medium in a two dimensional polar coordinate grid. Discussion includes solving the mass balance equation, establishing streamlines throughout the reservoir medium and calculating flow rates via a stream tube approach. Several simplified cases are discussed to establish the validity of the model.

Chapter four discusses application of the model established in Chapter three to realistic scenarios. Three cases are investigated: a six factor cased, perforated and damaged well completion; a three factor cased, perforated and damaged well completion; and a three factor open-hole filer cake scenario. The importance of the various parameters influencing the models is determined using design of experiments methodology. An IV-

optimal response surface design is utilized to capture the nonlinear behavior of the models under study.

Chapter five provides an explanation of the modifications required to model streamlines in a three dimensional cylindrical coordinate system. Explanation is provided for both the isotropic and anisotropic cases in a heterogeneous medium. Appropriate modifications to the Laplace equation, streamline procedure and flow rate calculations are discussed.

Chapter six summarizes this thesis with conclusions as a result of the research performed, a summary of the novel aspects within the research and various recommendations for future work.

- Chapter 2 -

2.0 Literature Review

2.1 Introduction

The flow of fluids through porous media has been studied for over 150 years. Henry Darcy is credited as the innovator in this field of physics, establishing what he referred to as the law of fluid mechanics (Ritzi Jr. & Bobeck, 2008). Jules Dupuit, a colleague of Darcy, was also a highly respected researcher studying the physics of groundwater flow. Together, the cumulative work of Darcy (1856) and Dupuit (1857) established numerous quantitative hydrogeological laws, including the governing equations for groundwater motion, average linear velocities, average travel time, hydraulic conductivity in heterogeneous media, conservation of mass in confined and unconfined flow, the nature of pieziometric surfaces, dual porosity flow and various other significant physical phenomena (Ritzi Jr. & Bobeck, 2008).

Since these original works much knowledge has been gained towards the understanding of fluid flow through porous media. Within the petroleum field, understanding of additional complexities involving compressible fluids, phase behavior and fluid-rock interaction has been the focus of extensive research. Summaries of such research are found in the work of Scheidegger (1960, 1974) and Bear (1972, 1988).

The focus of this research deals with streamline modeling. The remainder of this chapter deals with the development of the current approach to streamline modeling, its predecessor methodologies and the areas of study in modern research.

2.2 Streamline Modeling

2.2.1 Streamline and Stream Tube Modeling Origins

The first use of streamlines dates back to the work of Muskat (1937). Muskat's work dealt specifically with groundwater flow through porous media. With a broad scope, aspects of his work involved modeling fluid flow and transport via streamline related procedures.

Application and growth of the underlying concepts developed by Muskat have been undertaken in various aspects of the petroleum reservoir field since his original work (Datta-Gupta & King, 2007). The first instances of streamline modeling within the petroleum field come from the work of Fay and Prats (1951) who modeled two dimensional flow fields for use in determining invasion patterns in cycling and flooding problems (Fay & Prats, 1951). Numerical methods were utilized to model flow for a single fluid and dual fluid system, producing increased accuracy in breakthrough time for a waterflood.

Higgins and Leighton (1961) introduced a new analytical procedure to model two phase flow in complex rock geometry. Their procedure uses a two dimensional stream tube approach to model a two phase displacement, with a discussion of the potential for a three phase approach. As with the previous methodologies, the procedure involved dividing the flow into a number of stream tubes and performing fluid saturation calculations along the tubes. The method was for water and oil, working in a "fast and accurate" manner for reservoirs with complicated well spacings (Higgins & Leighton, 1962).

Morel-Seytoux (1966) studied the influence of pattern geometry on oil recovery for a specific displacement mechanism. Various assumptions were applied to the specific model, including a unit mobility ratio for two phase flow, piston like front displacement, incompressible flow and no gravitational or capillary pressure effects. The work provides a concise and simple solution to the proposed problem using stream tube models, and states that its true value will be in the future application of the technique for non-unit mobility ratio scenarios, leading to a more realistic solution (Morel-Seytoux, 1966).

Pitts and Crawford (1970) were the first to introduce permeability heterogeneity into the streamline modeling process. By allowing permeability to vary in a realistic manner throughout the reservoir and accommodating via appropriate modeling modifications, they showed reduced areal sweep efficiency for heterogeneous media versus homogenous media (Pitts & Crawford, 1970). This is as a result of fluid flowing around tight zones and flowing more easily through high permeability areas. The procedure lead to an improvement in accuracy compared to that of Muskat's initial methodology, but still encompasses the same assumptions of incompressibility, negligible gravity and capillary pressure effects and two dimensional flow.

LeBlanc and Claude (1971) produced a streamline model capable of handling two phase flow, variable mobility ratios and reservoir heterogeneity. The model was best suited for secondary recovery predictions for reservoirs with arbitrary well placements undergoing the aforementioned sweep and reservoir characteristics. Unlike previous methodologies, the work of LeBlanc and Claude was the first to focus solely on streamlines as oppose to stream tubes or stream bundles. This is favored since no knowledge of the geometric shape is required for volumetric flow calculations (LeBlanc & Caudle, 1971). This form

of streamline modeling was especially valuable for water floods in which the slowly changing velocity field leads to reduced computational time and a lack of numerical dispersion (Datta-Gupta & King, 2007). The assumptions of incompressibility, negligible gravitational and capillary pressure effects and two dimensional flow are employed in this method.

Martin and Wegner (1979) performed further work by establishing a numerical method to overcome the effects of numerical dispersion. This work focused on stream tube modeling, studying the effects of fixed stream tubes versus variable stream tubes over a range of mobility ratios. A numerical solution for two-phase, two dimensional incompressible flow was achieved by solving Buckley-Leverett equations in stream tubes and repeating the calculation as the flood progressed (Martin & Wegner, 1979). The effects of capillary pressure and gravity were neglected.

2.2.2 Hybrid Approach to Three Dimensional Modeling

The first attempt to apply streamlines in three dimensions was performed by Lake et al. (1981) in the modeling of a large scale surfactant/polymer project. Direct attempts to model streamlines in three dimensions were difficult due to complexities involved with tracking tube geometries (Datta-Gupta & King, 2007). Lake applied a hybrid approach by combining a cross-sectional finite difference simulator with an areal stream tube model resulting in the first three dimensional use of streamlines (Lake et al., 1981). The technique was an approximation, but represented a more efficient approach than the independent use of both modeling types.

Numerous successful applications of the technique proposed by Lake et al. (1981) are available in the literature. Emanuel et al. (1989) used a similar approach in developing a systematic method to blend fractal statistics, detailed geological data, finite difference simulation and stream tube modeling to improve reservoir performance predictions (Emanuel et al., 1989). The incorporation of a cross-sectional finite difference simulation allowed for the modeling of more complex fluid behavior, including miscibility and three phase flow, while the areal stream tube model allowed for much quicker processing speed.

The technique was also used in the work of Mathews et al. (1989) for the modeling of a miscible water-alternating-gas (WAG) flood. The work compared the modeling procedures of the fractal hybrid approach to the conventional modeling processes. The results indicated that the finite difference/stream tube hybrid technique based on fractal representations can be an adequate modeling technique, providing an efficient use of effort and increased computational speed (Mathews et al., 1989).

In the work of Emanuel and Miliken (1997) a comparison is made between a three dimensional finite difference model and a hybrid stream tube/finite difference model used for full field waterflood simulation. The work notes the many computational advantages of three dimensional finite difference simulators, including the ability to model complex physics uncaptured within the stream tube approach. But it also recognizes the uncertainties in the modeling process, as well as the computational time restraints. The stream tube/finite difference hybrid approach offers enhanced flow visualization, allowing for a greater understanding of the nature of flow throughout the reservoir, in addition to less computational requirements.

2.2.3 Front Tracking

Front tracking is a method of calculating convective transport of particles throughout a reservoir medium. A front is a curve (two dimensions), or a surface (three dimensions), connecting particles having the same respective properties. It may be viewed as contours connecting streamlines. The fronts are drawn either for equal time of flights or equal concentrations, often at discontinuities or shocks during flooding processes. The progression of these fronts is the focus of front tracking simulation.

The front tracking methodology was initially established by Glimm et al. (1981, 1983). Using hyperbolic systems of equations Glimm modeled discontinuous solutions using fine and coarse grid resolutions. The work established a means to track the progression of shock fronts with no numerical dispersion (Glimm et al., 1981). The initial models were proven successful for homogeneous and heterogeneous reservoir mediums, as well as variable mobility ratios and immiscible displacements.

Bratvedt et al. (1992) developed a new front tracking method for reservoir simulation. The method is based upon the initial theory proposed by Glimm, but takes advantage of the new developments within the field of hyperbolic conservation laws (Bratvedt et al., 1992). Using these developments, a new method was generated for solving saturation equations throughout the modeling process. The method does not exhibit any numerical dispersion, is not altered by grid related effects and is computationally more efficient than finite difference models (Bratvedt et al., 1992).

2.2.4 Lagrangian Particle Tracking

Lagrangian particle tracking is another method for calculating convective transport of particles throughout a reservoir medium. This method, developed by Schafer-Perini and Wilson (1991), focuses on statistically significant concentrations of particles. The particles are representative of fluids within the reservoir. Movement of the particles is determined by calculating the velocities throughout the reservoir medium and moving the particles along the resultant pathlines. Numerical dispersion is then accounted for via a specific algorithm. After convective steps, the particles are redistributed for dispersion in proportion to the calculated variance (Datta-Gupta & King, 2007).

The Lagrangian particle tracking method works well for floods progressing with sharp fronts, but is inefficient for smooth fronts experiencing large dispersion. In addition, the method tends to force the fronts to lose resolution with time as the dispersion is related to the variance of the concentration (Datta-Gupta & King, 2007).

2.2.5 Level Set Methods

Level set methods are another form of convective transport modeling dealing with progression of interfaces through porous media. Level set methods represent an alternative to previous front tracking procedures in the mathematical basis for tracking the fronts.

The methods were first developed by Osher and Sethian (1988). They use numerical algorithms based on Hamilton-Jacobi formulations to track front propagation with curvature dependent speed (Osher & Sethian, 1988). The front propagation is considered the zero level set of a higher dimensional function, with the propagation of the front

being solved via a partial differential equation (Datta-Gupta & King, 2007). Further details on the mathematics involved within this method may be found in the work of Sethian (1996).

2.2.6 Modern Approach

An innovation in the approach to streamline modeling was developed by Pollock (1988). Pollock determined a semi-analytical approach to particle tracking based upon grid boundary velocities known through the solution of finite difference models. Using the known face velocities along each grid cell, linear interpolation is used to determine the velocities throughout the entire medium. Using this assumption and a known entrance point, the path of a particle may be traced utilizing a time of flight calculation, followed by recording the coordinates from grid face to grid face. Pollock's method is extremely simple and accurate. In addition, it can account for no flow boundary conditions as well as transient flows with the inclusion of time steps.

One of the earliest extensions of Pollock's particle tracking methodology in the reservoir engineering field was by Datta-Gupta and King (1995). In this research the particle tracking algorithm was applied to a heterogeneous reservoir medium with an arbitrary configuration of wells. The approach is used to study tracer transport throughout the reservoir medium. The research also presents the use of time of flight as a spatial variable. This approach is first suggested in a paper by Johansen (1992). Johansen proposed an extension of simple wave theory to stream tube modeling. The research states that stream tube models involving two space dimensions may be reformulated to a single dimensional problem along individual stream tubes. This is performed by introducing dimensionless variables (dimensionless distance and dimensionless time).

The methods proposed by Datta-Gupta and King (1995) are similar to that of Johansen (1992) with the exception of formulating the problem along individual streamlines as oppose to stream tubes. In both cases transport and saturation calculations can be reformulated from three dimensional problems to one dimensional problems along streamlines or stream tubes. As a result, heterogeneities and the finite difference grid become decoupled from the transport calculations. The additional assumptions in the methods of Datta-Gupta and King (1995) eliminate the need for evaluating stream tube geometries.

The modern streamline approach has been studied and researched by various authors, with many new advances being contributed to the field of study. Bratvedt et al. (1996) first modeled streamlines in three dimensions via an extension of the Pollock particle tracking methodology. In addition, the effect of gravity was implemented via operator splitting (Bratvedt et al., 1996). The approach of Bratvedt et al. utilized front tracking along streamlines as well as the transformation to one dimensional saturation calculations along streamlines to simplify problems to a single dimension. It is noted to be accurate and computationally efficient when compared to equivalent transport calculations within finite difference simulators.

The extension to unsteady state flow was first described by Thiele et al. (1996). In this research streamline modeling is used for convection dominated flow in highly heterogeneous reservoirs (neglecting gravity and capillary pressure). The novelty of the work is in the temporal updates to the streamlines, allowing for modeling of unsteady flow. The research utilized the stream tube approach for two dimensional problems and the streamline approach involving the one dimensional transformation for three

dimensional problems. Phenomena included within its scope consist of tracer flow, two phase immiscible displacement, first contact miscible displacement and a special case involving a million grid blocks. It notes that the speed and robustness of the techniques illustrated make it ideal for statistical reservoir forecasting and uncertainty evaluation.

Batycky et al. (1997) extended the streamline approach to full field simulation. Their method could account for changing well conditions, heterogeneity, mobility and gravitational effects (Batycky et al., 1997). Again, fluid transport calculations are performed along streamlines, simplifying the saturation calculations to a single dimension. Gravity is accounted for using an operator splitting technique. In addition, the streamlines are allowed to dynamically change as required by changes within mobility and boundary conditions (Batycky et al., 1997). The technique is used to successfully model a full field reservoir undergoing a waterflood with 36 wells in total.

Peddibhotla et al. (1996) used streamline modeling techniques to evaluate multiple fine scale realizations of heterogeneous geological models. Traditionally, an upscaling technique is used to create a coarse grid model for subsequent flow modeling. Peddibhotla et al. performed flow calculations utilizing geo-statistical models, allowing for the evaluation of multiple realizations and a greater quantification of the uncertainty in predicting dynamic reservoir fluid behavior (Peddibhotla et al., 1996).

Peddibhotla et al. (1997) used streamline modeling in field scale applications with increasing accuracy. Novelties of the approach include using a three dimensional mapping algorithm during changing well conditions and utilizing a third order total

variation diminishing method to handle multiphase flow and minimize numerical dispersion (Peddibhotla et al., 1997).

Ingebrigsten (1999) presented two solutions for calculations involving three phase compressible flow in porous media using streamlines. Up to this point streamline modeling neglected compressibility due to the strong coupling between pressure and saturation (Ingebrigsten et al., 1999). Ingebrigsten et al. present both a sequential method involving additional pressure updates and an implicit method solving for both pressure and saturation simultaneously. The authors note that the sequential method performs well in comparison with finite difference simulation, while the implicit method had not been tested to a rigorous level.

A paper produced by Thiele (2001) summarizes the developments and capabilities of streamline modeling up to the date of publication. It outlines what the author considers to be the six key principles in streamline modeling:

- Three dimensional particle tracking based upon time of flight;
- Transforming mass conservations equations along streamlines in terms of time of flight;
- Periodically update streamlines to account for changing conditions;
- Numerical one dimensional transport calculations along streamlines;
- Accounting for gravity via operator splitting; and
- Extension to compressible flow.

The paper also outlines the various uses of streamline modeling within the area of reservoir simulation. These include upscaling, quantifying displacement efficiency,

history matching and field optimization (Thiele, 2001). Thiele credits the success of streamline simulation to its clear flow visualization, its capability for full field modeling, its efficiency, its computational speed and its increasing ability to model more complicated physics.

2.2.7 Recent Research

In the last 10 years various advancements have been made in the field of streamline modeling. The advancements span a wide range of topics, including compositional simulation, tracking streamlines through complex reservoirs, parallel simulation, history matching and well testing applications. The list of authors and contributions are numerous, with excellent research coming from the likes of Akhil Datta-Gupta at Texas A&M University, Marco Thiele and Rod Batycky at Stanford University, and numerous other authors worldwide. Much of the advancements are in areas outside the scope of the research presented in this thesis and therefore will not be reviewed in detail. The remainder of this chapter highlights a selection of the significant papers that have been produced over the last 10 years.

Datta-Gupta (2000) produced a paper reviewing the current status of streamline modeling in the field of reservoir simulation. The paper presents a review of its history, formulation, current applications and potential limitations. Noted benefits include flow visualization, well allocation factors, dynamic reservoir characterization and flood front management (Datta-Gupta, 2000). Noted limitations include the inability to model complex physics and cross streamline mechanics.

The work of He et al. (2002) details how streamline modeling may be used to identify reservoir compartmentalization and flow barriers during primary production. The approach works via a three step method. First, traditional decline curve analysis is performed on production data to determine well communication and individual well drainage volumes. Next, streamline models are created utilizing existing geological models to identify a new set of drainage volumes. In the final step, reservoir compartmentalization and flow barriers are determined via matching the drainage volumes from the decline curve analysis with the streamline drainage volumes. The approach is best suited for primary production with a limited number of wells (He et al., 2002). The approach is validated with a field application in the Gulf of Mexico.

A paper written by Matringe and Gerritsen (2004) investigates the accuracy of streamline modeling. The focus is on the tracing of the individual streamlines and their approximation to reality. Since the true analytical solution is known for a homogenous quarter five spot pattern, Matringe and Gerritsen ran various simulations using this setup. Their results show that numerous errors may result when using streamline models, including errors in location, arc length and time of flight. The errors are contributed to inaccuracies in flux values at grid block faces, streamline starting locations and low order interpolation between grid blocks (Matringe & Gerritsen, 2004). They also show various methods to improve streamline modeling accuracy, including the use of mixed hybrid finite element methods as oppose to finite difference methods to improve the accuracy of the fluxes, selecting proper starting locations to reduce tracing error and using adaptive mesh refinement or second order interpolation to improve the velocity distribution calculations within grid cells (Matringe & Gerritsen, 2004).

A paper produced by Kippe et al. (2007) looks to improve upon the mass balance in streamline methods. As discussed previously, streamlines are first generated on a static finite difference grid solving for pressures and velocities. Following this, a transformation is performed for fluid transport calculations reducing the three dimensional problem to a single dimension along streamlines. This is known as a sequential Eulerian-Lagrangian procedure. The mapping of saturations using this procedure and the back and forth transformation results in mass balance errors that can accumulate to a significant level with time. The research of Kippe et al. develops a new mapping algorithm designed to improve the mass conservation procedure while minimizing computational burden. This is achieved by changing quantities in the transport equation locally, resulting in increased accuracy and maintaining computational efficiency (Kippe et al., 2007).

History matching is an area in which streamline simulation is proving valuable. The process involves starting with a geological model and simulating flow, with an end goal of a simulated production profile equivalent to the actual field production history to date. A paper published by Cheng et al. (2007) develops a methodology for three phase history matching using compressible streamlines. Streamline approaches previous to this mainly dealt with two phase oil-water flow assuming incompressible or slightly compressible flow. The method works by utilizing a density dependent source term in the saturation equation and introducing an effective density along the streamlines. This allows solutions to remain in the single dimensional form along with accounting for pressure effects during saturation calculations (Cheng et al., 2007). In addition, analytical calculations are used to determine the relationship between reservoir properties and the production

response. The history matching approach is tested using synthetic cases and field scale experiments, proving the method to be significantly faster than the traditional judgment based trial and error approach.

A paper produced by Jimenez et al. (2008) studies full field streamline tracking in complex faulted systems containing non-neighboring connections between grid cells. In modern streamline tracking a variety of cell geometries may be used spanning simple rectangles to complex corner point designs. In finite difference simulators, juxtaposition along faults may cause cells to have several non-neighboring connections. Within the simulators, flux to each of these cells is calculated individually. Utilizing the traditional Pollock methodology for streamline tracing in these regions will not accurately represent the flux since the distribution may not be uniform. The work presented by Jimenez et al. presents a simple concept for proper representation of streamlines in this vicinity. The approach utilizes localized grid refinement along faulted cells. This allows the fluxes to be accurately honored across non-neighboring grid cells. The concept of local refinement to enhance streamline accuracy is elaborated throughout the research to be presented within this thesis.

Recent work by Nilsen and Lie (2009) details the use of front tracking methodologies for compressible streamlines. Taking advantage of the new advancements in compressibility calculations along streamlines, Nilsen and Lie implement a front tracking methodology for compressible two phase flow. The work is performed on three dimensional full field models. Their results indicate accurate simulation of compressible flow can be performed utilizing both streamline modeling and front tracking methodologies (Nilsen & Lie, 2009).

Recent work by Kumar et al. (2009) shows the use of phase and component streamline tracking in understanding various reservoir mechanisms. As opposed to using total flux, the method here looks to trace phase and component flux. In doing so, the ability for flow simulation is lost due to the discontinuous nature of these fluxes, but additional insight is gained into the nature of processes and mechanisms occurring within the reservoir. The technique is used for both synthetic cases and full field applications. Some of the benefits of this approach include identifying regions in which different drive mechanisms are active, knowing the dominant flowing phases in different locations within the reservoir and using the information to optimize future infill well locations (Kumar et al., 2009). The method is also performed on a CO_2 injection project in Canada, providing the ability to distinguish between CO_2 in the gas and liquid phase, helping to optimize injection and providing a method to evaluate the CO_2 sequestration effectiveness (Kumar et al., 2009).

A recent piece of literature concerning streamline modeling has been produced by Thiele et al. (2010). Similar to Thiele's paper in 2001, the work is mostly a review of the streamline modeling process to date, describing how the general modeling process works and highlighting areas in which it has proven to be valuable. These include reservoir-flow surveillance, flow simulation, history matching and flood management (Thiele et al., 2010). The work presents both the advantages and disadvantages of streamline modeling, and notes that "whether streamline simulation can be of benefit depends strongly on the questions being asked of the model, the assumptions engineers are willing to make and ultimately the time available for a reservoir study" (Thiele et al., 2010). The paper also looks into the future of streamline modeling, suggesting that significant benefit may come from a functional tool incorporating both finite difference and streamline

simulation models. This hybrid model could allow a user to switch between modeling options, or have both working automatically, creating a fit for purpose tool for future reservoir simulation (Thiele et al., 2010).

2.3 Additional Works of Interest: Skin and Perforation Effectiveness

The research presented in this thesis focuses on streamline modeling within the near wellbore region of an oil well. The nature of heterogeneities surrounding the wellbore and their effect on production is an essential aspect of this research. In addition, the modeling of well completions and their overall effect on flow is another area of focus. All of the parameters which influence flow within the near wellbore region may be conveniently modeled as skin. A paper produced by Yildiz (2006) provides an assessment of total skin factors in perforated wells. The work provides a comprehensive review of all available methods and software used to predict well productivity and total skin factors in fully perforated vertical wells (Yildiz, 2006). The paper also provides a detailed description of the various components which contribute to skin.

A paper by Karakas and Tariq (1991) reviewed the effect of perforations and various reservoir parameters on the productivity of a well. Semi-analytical models are developed to predict skin effects as a result of the perforations and near wellbore parameters. Their method for predicting mechanical skin is widely accepted within the petroleum industry. It is noted for its simplicity, ease of use and theoretical foundation (Kabir et al., 2009).

A paper by Kabir et al. (2009) provides a review of the total mechanical skin model developed by Karakas and Tariq (1991). The research notes that while the Karakas and Tariq model is effective for perforations that do not extend beyond the damage zone, it is ineffective and illogical for perforations extending beyond this reach. The authors propose an analytical model which honors the Karakas and Tariq model when perforations stop within the damage zone, but delivers more accurate results when perforations extend beyond this region. The method separates flow into two regimes;

fluid that flows through the damage zone and fluid that flows directly into the perforations. Their results are said to be encouraging and ready for inflow software implementation.

Numerous other studies have been performed to determine the effect of various completion strategies on well productivity. A paper produced by Shedid and Zekri (2006) looks to determine the effects of fractions of perforated length, wellbore length and associated reservoir fractures on the overall productivity of horizontal wells. Experimental procedures are used to determine the results. A paper produced by Ibrahim et al. (2009) reviews a novel perforating technique to improve wellbore performance. The papers provide a review of the various perforation characteristics which may influence a well production, including gun size, charge type, shot density, reservoir characteristics and well conditions. The paper also outlines the importance of the cleanup phase and its effect on well productivity.

- Chapter 3 -

3.0 Streamline Modeling

3.1 Introduction

As mentioned in Chapter 1, streamlines are a set of curves that are tangent to the local velocity field at a given instant in time (Thiele et al., 2010). As a result, streamline models show the direction a fluid element will travel at a given time instant. For a steady state system, streamlines are equivalent to pathlines. Pathlines show the trajectory of particles throughout a given region, in this case the reservoir. For a steady state scenario the velocity field and pressure distribution will not change with time. As a result the streamlines will remain constant and equal to the pathlines. If pressure is allowed to change with time an unsteady state condition exists. As the pressure distribution changes, so to must the velocity field, and hence the streamlines are altered. In this case streamlines will not align with pathlines as the streamlines represent the instantaneous movement, while pathlines are influenced by the changes in streamlines with time.

In a general sense, streamlines provided an instantaneous view of the fluid flow characteristics throughout a reservoir. For steady state flows, streamline models go beyond an instantaneous viewpoint, providing insight into the full nature of flow throughout a reservoir independent of time.

In this chapter the methodology for developing streamline models in a polar geometry is discussed. The methodology presented is an expanded version of that derived by Johansen (2009). This includes establishing the model assumptions, the intricacies of

dealing with polar coordinates, solving the Laplace equation, generating streamline models and determining flow rate. Several aspects of the methodology are novel, presenting improvements and correction upon current techniques. Simplified case studies are provided to verify the approach and enhance the understanding of the modeling procedure. The remaining chapters deal with establishing realistic results and modeling modifications, all of which originate from the procedures established in this chapter.

3.2 Model Assumptions

Several underlying assumptions are maintained throughout the modeling process. These range from the nature of gridding to the dominant flow phenomena throughout the reservoir medium.

The streamline models created throughout this research are derived for use in a polar coordinate grid. The model has a well located at the center of the polar grid with Cartesian coordinates $(x, y) = (0,0)$ translating into polar coordinates of $(r, \theta) = (0,0)$. The well has an exterior radius of r_w with a corresponding constant pressure of p_{wf}. Likewise, the exterior boundary of the reservoir is r_e with a constant external pressure of p_e. The reservoir has a constant thickness represented by h. Figure 16 provides a schematic of the well and reservoir geometry.

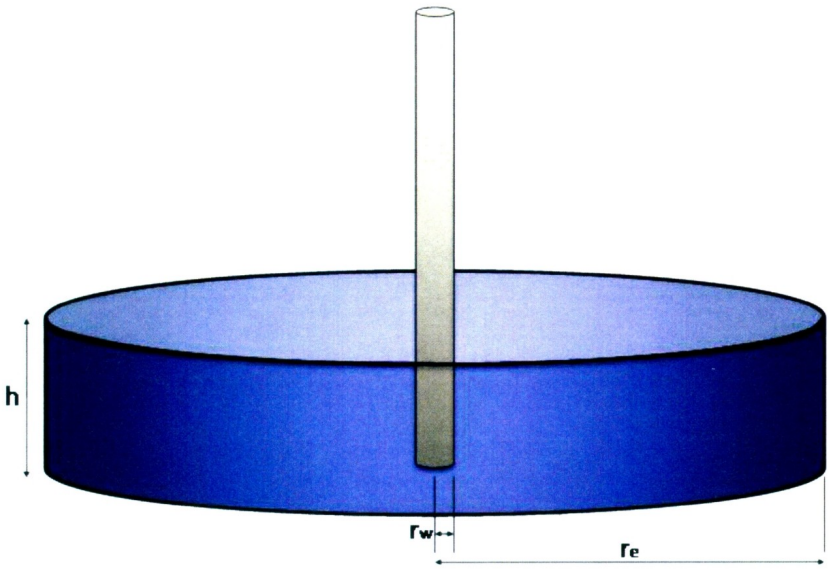


Figure 16 - Schematic of Wellbore and Reservoir Geometry

A constant pressure at both the internal and external boundaries implies steady state flow. In steady state flow, flow rate is independent of time, resulting in a constant flow rate.

The scenario is realistic for large reservoirs in the early stages of production, as well as reservoirs with maintained pressure support via strong aquifers or water injection.

The difference in pressure between the two boundaries is the primary driving force for the flow of oil throughout the reservoir medium. The nature and scale of the flow will also be affected by various reservoir parameters. These parameters are discussed throughout this section.

Streamline models are based upon differential pressure being the primary drive mechanism. Flow phenomena relying on other parameters such as absolute pressure or gravitational flow are currently not well suited for streamline models. The potential driving effects not incorporated into this modeling process includes miscibility, capillary pressure and gravity. Worldwide, several researchers are working at ways to incorporate these effects, currently modeling via finite difference simulations, within streamline models to provide the best overall solution to the complex reservoir modeling task (Thiele et al., 2010).

Another important assumption of this modeling procedure is incompressible flow. The compressibility of oil, along with numerous other physical and chemical properties, is a strong function of pressure, temperature and composition. This dependence results in various complex phenomena, including fluid behavior, phase behavior and fluid-rock interactions. For the purpose of this modeling the dependence is ignored, allowing for the modeling of an incompressible fluid with constant viscosity. The assumption is realistic for an under-saturated reservoir in which the wellbore pressure remains above the bubble point pressure of the fluid, resulting in a single phase flow of oil throughout the reservoir

medium. Several researchers have performed work in regards to compressible streamlines, but this is currently beyond the scope of this research.

The streamline modeling carried out in this research is performed in two dimensions. The reservoir is viewed as a cylinder with a constant thickness. The well is perpendicular to the reservoir and parallel to the z-axis. All streamlines are assumed to move parallel to each other in the vertical plane and perpendicular to the axis of the wellbore. This is realistic for a well completed the full thickness of the reservoir to prevent vertical movement in the streamlines and maintain radial flow with respect to the vertical direction. The extension to three dimensional modeling is discussed in Chapter 5.

The variation in permeability throughout the reservoir medium is the primary variable of interest in this research. In a reservoir medium permeability may vary in each direction. In a Cartesian coordinate system this would be represented by K_x, K_y and K_z. Likewise, in a radial coordinate system these would be translated to K_r, K_t and K_z. Permeability dependent upon direction is known as anisotropy. If the permeability is the same in each direction, the reservoir is considered an isotropic medium. In practice, most reservoirs have some degree of anisotropic conditions, but near isotropic reservoirs do exist.

In addition to directional variation, permeability may also vary spatially, becoming a function of location. A homogenous reservoir medium is one in which permeability is constant spatially, independent of position. If the reservoir permeability varies with location it is a heterogeneous reservoir. Most reservoirs are heterogeneous, with permeability varying throughout the reservoir medium. This is especially evident in the

near wellbore region, in which various effects from drilling and completions alter the permeability distribution in this vicinity.

This chapter describes the modeling process for a heterogeneous isotropic reservoir. Some discussion is also provided for the most complex scenario involving heterogeneous and anisotropic conditions.

In the modeling process it is important to align the Cartesian coordinate system with the principle permeability directions K_x, K_y and K_z. In three dimensions, the permeability of an anisotropic reservoir can be represented by the following permeability tensor:

$$K = \begin{pmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{xy} & K_{yy} & K_{yz} \\ K_{xz} & K_{zy} & K_{zz} \end{pmatrix}$$

It is a mathematical fact that three directions perpendicular to each other can be found which reduces the permeability tensor to a diagonal matrix (Johansen, 2008). These are the principle permeability directions which correspond to the initial laying of the sandstones millions of years ago. The permeability matrix becomes:

$$K = \begin{pmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{pmatrix}$$

Neglecting the vertical direction for two dimensional modeling, this reduces to:

$$K = \begin{pmatrix} K_x & 0 \\ 0 & K_y \end{pmatrix}$$

The initial grid is created such that the x, y and z axis correspond with the principle permeability directions (K_x, K_y and K_z).

3.3 Reservoir Modeling in Polar Coordinates

As stated previously, the streamline models generated throughout this research are performed entirely in a polar coordinate system. Therefore it is important to understand how to convert from the traditional (x, y) Cartesian coordinates to the (r, θ) polar coordinates. The trigonometric relationships between these two systems are as follows:

$$x = r\cos\theta \qquad y = r\sin\theta$$

$$r = \sqrt{x^2 + y^2} \qquad \theta = \arctan\left(\frac{y}{x}\right)$$

Figure 17 provides a schematic for the variables in the relations provided above. Note that the radius is measured from the $(0,0)$ mark on the Cartesian coordinate system, while the angle θ is considered zero on the positive x-axis and has a positive increase in the counterclockwise direction.

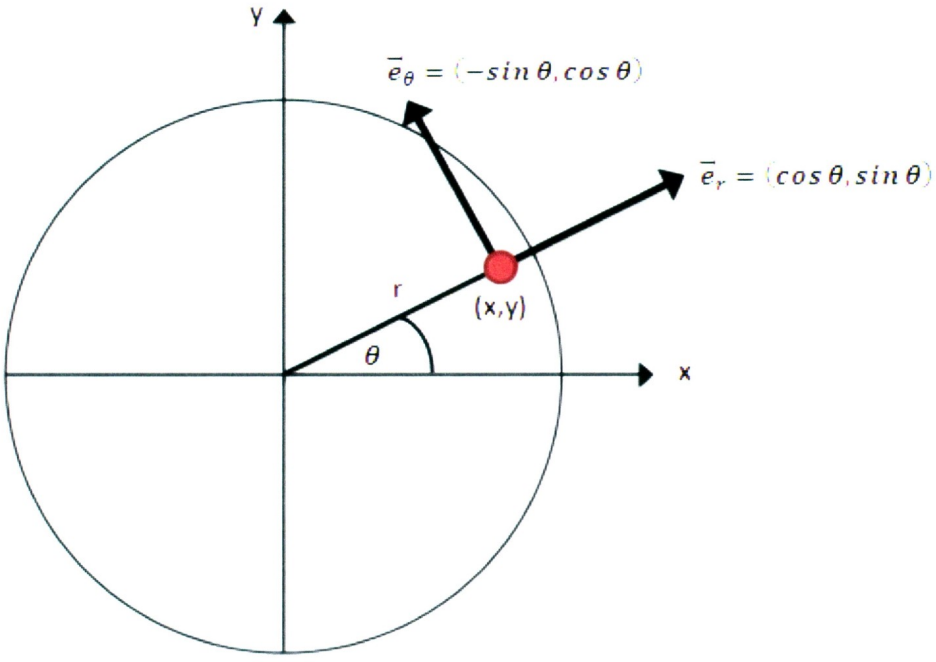


Figure 17 - Notation in Radial Geometry

Also provided in Figure 17 is the notation of the unit vectors. At any point in the polar system two unit vectors aligned with the polar coordinate system will always exist. Here, $\vec{e}_r$ is the unit vector in the radial direction and is represented by $\vec{e}_r = (\cos\theta, \sin\theta)$. Ninety degrees perpendicular to this is the second unit vector $\vec{e}_\theta$, represented by $\vec{e}_\theta = (-\sin\theta, \cos\theta)$. Recall that the magnitude or length of a unit vector is always equal to one.

The gradient of a vector creates a new vector in which the components are the partial derivatives of the original vector. In Cartesian coordinates this is represented as:

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$$

Applying the chain rule for differentiation in the x-direction:

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial}{\partial \theta}\frac{\partial \theta}{\partial x} = \frac{\partial}{\partial r}\frac{\partial}{\partial x}\left(\sqrt{x^2 + y^2}\right) + \frac{\partial}{\partial \theta}\frac{\partial}{\partial x}\left(\arctan\left(\frac{y}{x}\right)\right)$$

which reduces to:

$$\frac{\partial}{\partial x} = \cos\theta\frac{\partial}{\partial r} - \frac{1}{r}\sin\theta\frac{\partial}{\partial \theta}$$

Likewise in the y-direction:

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial}{\partial \theta}\frac{\partial \theta}{\partial y} = \frac{\partial}{\partial r}\frac{\partial}{\partial y}\left(\sqrt{x^2 + y^2}\right) + \frac{\partial}{\partial \theta}\frac{\partial}{\partial y}\left(\arctan\left(\frac{y}{x}\right)\right)$$

which reduces to:

$$\frac{\partial}{\partial y} = \sin\theta\frac{\partial}{\partial r} + \frac{1}{r}\cos\theta\frac{\partial}{\partial \theta}$$

Therefore the gradient in polar coordinates becomes:

$$\nabla = \left(\cos\theta \frac{\partial}{\partial r} - \frac{1}{r}\sin\theta \frac{\partial}{\partial \theta}, \sin\theta \frac{\partial}{\partial r} + \frac{1}{r}\cos\theta \frac{\partial}{\partial \theta}\right)$$

3.3.1 Polar Vector Formulation

To further convert the formulation in polar coordinates the components of the gradient in the direction of the unit vectors must be determined. This is performed by taking the dot product of the gradient and the unit vectors individually. For the $\vec{e}_r$ direction:

$$\nabla_r = \vec{e}_r \cdot \nabla$$

$$\nabla_r = (\cos\theta, \sin\theta) \cdot \left(\cos\theta \frac{\partial}{\partial r} - \frac{1}{r}\sin\theta \frac{\partial}{\partial \theta}, \sin\theta \frac{\partial}{\partial r} + \frac{1}{r}\cos\theta \frac{\partial}{\partial \theta}\right)$$

$$\nabla_r = \cos\theta \left(\cos\theta \frac{\partial}{\partial r} - \frac{1}{r}\sin\theta \frac{\partial}{\partial \theta}\right) + \sin\theta \left(\sin\theta \frac{\partial}{\partial r} + \frac{1}{r}\cos\theta \frac{\partial}{\partial \theta}\right)$$

which reduces to:

$$\nabla_r = \cos^2\theta \frac{\partial}{\partial r} + \sin^2\theta \frac{\partial}{\partial r} = \frac{\partial}{\partial r}$$

For the $\vec{e}_\theta$ direction:

$$\nabla_\theta = \vec{e}_\theta \cdot \nabla$$

$$\nabla_\theta = (-\sin\theta, \cos\theta) \cdot \left(\cos\theta \frac{\partial}{\partial r} - \frac{1}{r}\sin\theta \frac{\partial}{\partial \theta}, \sin\theta \frac{\partial}{\partial r} + \frac{1}{r}\cos\theta \frac{\partial}{\partial \theta}\right)$$

$$\nabla_\theta = -\sin\theta \left(\cos\theta \frac{\partial}{\partial r} - \frac{1}{r}\sin\theta \frac{\partial}{\partial \theta}\right) + \cos\theta \left(\sin\theta \frac{\partial}{\partial r} + \frac{1}{r}\cos\theta \frac{\partial}{\partial \theta}\right)$$

which reduces to:

$$\nabla_\theta = \frac{1}{r}\sin^2\theta\frac{\partial}{\partial\theta} + \frac{1}{r}\cos^2\theta\frac{\partial}{\partial\theta} = \frac{1}{r}\frac{\partial}{\partial\theta}$$

Therefore the gradient in polar coordinates can be expressed as:

$$\nabla_{r,\theta} = \left(\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial\theta}\right)$$

The flux vector $\vec{u}$ represents the volumetric flux or Darcy velocity for flow throughout the porous medium. It is calculated by dividing the total flow rate by the cross-sectional area in a given direction. The divergence of the flux vector $\vec{u}$ in Cartesian coordinates is:

$$\nabla \cdot \vec{u} = \left(\frac{\partial u_x}{\partial x}, \frac{\partial u_y}{\partial y}\right)$$

To express the divergence of the flux vector in polar coordinates the divergence along the unit vectors $\vec{e}_r$ and $\vec{e}_\theta$ must be determined. In polar coordinates the flux vector is expressed as:

$$\vec{u} = (u_r, u_\theta)$$

For the following calculations it is important to know the differentials of the unit vectors with respect to the radial and tangential directions. These are as follows:

$$\frac{\partial\vec{e}_r}{\partial\theta} = \frac{\partial}{\partial\theta}(\cos\theta, \sin\theta) = (-\sin\theta, \cos\theta) = \vec{e}_\theta$$

$$\frac{\partial\vec{e}_\theta}{\partial\theta} = \frac{\partial}{\partial\theta}(-\sin\theta, \cos\theta) = (-\cos\theta, -\sin\theta) = -\vec{e}_r$$

$$\frac{\partial \vec{e}_r}{\partial r} = 0 \qquad \frac{\partial \vec{e}_\theta}{\partial r} = 0$$

It is also important to remember several mathematical principles concerning vector operations. First, the dot product of any unit vector and itself is equal to one. As well, the dot product of any perpendicular vector is equal to zero. It is also important to recall the product rule for differentiation, which is utilized throughout the derivation.

The divergence of the flux vector $\vec{u}$ in polar coordinates is calculated as follows:

$$\nabla_{r,\theta} \cdot \vec{u} = \left(\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial \theta}\right) \cdot (u_r, u_\theta)$$

$$\nabla_{r,\theta} \cdot \vec{u} = \left(\vec{e}_r \frac{\partial}{\partial r} + \frac{1}{r}\vec{e}_\theta \frac{\partial}{\partial \theta}\right) \cdot (u_r \vec{e}_r + u_\theta \vec{e}_\theta)$$

$$\nabla_{r,\theta} \cdot \vec{u} = \vec{e}_r \cdot \frac{\partial}{\partial r}(u_r \vec{e}_r) + \vec{e}_r \cdot \frac{\partial}{\partial r}(u_\theta \vec{e}_\theta) + \vec{e}_\theta \cdot \frac{1}{r}\frac{\partial}{\partial \theta}(u_r \vec{e}_r) + \vec{e}_\theta \cdot \frac{1}{r}\frac{\partial}{\partial \theta}(u_\theta \vec{e}_\theta)$$

$$\nabla_{r,\theta} \cdot \vec{u} = \vec{e}_r \cdot \vec{e}_r \frac{\partial}{\partial r}(u_r) + \vec{e}_r \cdot u_r \frac{\partial}{\partial r}(\vec{e}_r) + \vec{e}_r \cdot \vec{e}_\theta \frac{\partial}{\partial r}(u_\theta) + \vec{e}_r \cdot u_\theta \frac{\partial}{\partial r}(\vec{e}_\theta) + \frac{1}{r}\vec{e}_\theta$$
$$\cdot \vec{e}_r \frac{\partial}{\partial \theta}(u_r) + \frac{1}{r}\vec{e}_\theta \cdot u_r \frac{\partial}{\partial \theta}(\vec{e}_r) + \frac{1}{r}\vec{e}_\theta \cdot \vec{e}_\theta \frac{\partial}{\partial \theta}(u_\theta) + \frac{1}{r}\vec{e}_\theta \cdot u_\theta \frac{\partial}{\partial \theta}(\vec{e}_\theta)$$

$$\nabla_{r,\theta} \cdot \vec{u} = \frac{\partial}{\partial r}(u_r) + \frac{1}{r}\vec{e}_\theta \cdot u_r \vec{e}_\theta + \frac{1}{r}\frac{\partial}{\partial \theta}(u_\theta) - \frac{1}{r}\vec{e}_\theta \cdot u_\theta \vec{e}_r$$

$$\nabla_{r,\theta} \cdot \vec{u} = \frac{\partial}{\partial r}(u_r) + \frac{1}{r}u_r + \frac{1}{r}\frac{\partial}{\partial \theta}(u_\theta)$$

$$\nabla_{r,\theta} \cdot \vec{u} = \frac{1}{r}\frac{\partial}{\partial r}(r\, u_r) + \frac{1}{r}\frac{\partial}{\partial \theta}(u_\theta)$$

3.3.2 Darcy's Law

As discussed in Chapter 1 and 2, Darcy's law is the empirical result produced by Henry Darcy (1856). The law is the accepted procedure for determining laminar fluid flow through porous media. Darcy's law in Cartesian coordinates using the principle permeability directions in two dimensions, neglecting gravity, can be expressed as:

$$\vec{u} = -\frac{1}{\mu}\bar{K}\nabla p$$

$$\vec{u} = -\frac{1}{\mu}\begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix}\begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix}$$

To express Darcy's law in polar coordinates the dot product of the flux vectors $\vec{e}_r, \vec{e}_\theta$ with the unit vectors u_r, u_θ must be determined. In the radial direction this becomes:

$$u_r = \vec{e}_r \cdot \vec{u} = -\frac{1}{\mu}(\cos\theta\,, \sin\theta) \cdot \left(K_x \frac{\partial p}{\partial x}, K_y \frac{\partial p}{\partial y}\right)$$

$$u_r = -\frac{1}{\mu}(\cos\theta\,, \sin\theta) \cdot \left(K_x \cos\theta \frac{\partial p}{\partial r} - \frac{1}{r}K_x \sin\theta \frac{\partial p}{\partial \theta}, K_y \sin\theta \frac{\partial p}{\partial r} + \frac{1}{r}K_y \cos\theta \frac{\partial p}{\partial \theta}\right)$$

$$u_r = -\frac{1}{\mu}\left(K_x \cos^2\theta \frac{\partial p}{\partial r} - \frac{1}{r}K_x \sin\theta\cos\theta \frac{\partial p}{\partial \theta} + K_y \sin^2\theta \frac{\partial p}{\partial r} + \frac{1}{r}K_y \cos\theta \sin\theta \frac{\partial p}{\partial \theta}\right)$$

Likewise, in the tangential direction:

$$u_\theta = \vec{e}_\theta \cdot \vec{u} = -\frac{1}{\mu}(-\sin\theta\,, \cos\theta) \cdot \left(K_x \frac{\partial p}{\partial x}, K_y \frac{\partial p}{\partial y}\right)$$

$$u_\theta = -\frac{1}{\mu}(-\sin\theta\,, \cos\theta) \cdot \left(K_x \cos\theta \frac{\partial p}{\partial r} - \frac{1}{r}K_x \sin\theta \frac{\partial p}{\partial \theta}, K_y \sin\theta \frac{\partial p}{\partial r} + \frac{1}{r}K_y \cos\theta \frac{\partial p}{\partial \theta}\right)$$

$$u_\theta = -\frac{1}{\mu}\left(-K_x \cos\theta \sin\theta \frac{\partial p}{\partial r} + \frac{1}{r} K_x \sin^2\theta \frac{\partial p}{\partial \theta} + K_y \sin\theta \cos\theta \frac{\partial p}{\partial r} + \frac{1}{r} K_y \cos^2\theta \frac{\partial p}{\partial \theta}\right)$$

The concept of directional permeability now becomes important in this derivation. Several definitions of directional permeability exist as defined by the work of Scheidegger (1960). A single approach is used in this research, as described by Johansen (2008). Following the chosen derivation, the permeability in a given direction can be determined by:

$$K_n = \begin{pmatrix} n_x, n_y, n_z \end{pmatrix} \begin{pmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = n_x^2 K_x + n_y^2 K_y + n_z^2 K_z$$

where $n_x, n_y \& n_z$ are unit vectors in the $x, y \& z$ directions, respectively. Using the conventional Cartesian coordinate system and applying unit vectors measured from the respective axes this becomes:

$$K_n = K_x \cos^2\theta_x + K_y \cos^2\theta_y + K_z \cos^2\theta_z$$

This is the generalized formula for directional permeability in a given direction. Using this approach the following quantities for polar coordinates in two dimensions are determined:

$$K_r = K_x \cos^2\theta + K_y \sin^2\theta$$

$$K_t = K_x \sin^2\theta + K_y \cos^2\theta$$

$$K_\theta = \left(K_y - K_x\right) \sin\theta \cos\theta$$

Here K_r refers to the radial permeability, K_t refer to the permeability 90 degrees tangential to the radial direction, and K_θ is a mathematically required variable representing the difference between the two previous values. Applying these newly derived quantities to the polar flux vectors results in the following:

$$u_r = -\frac{1}{\mu}\left(K_r \frac{\partial p}{\partial r} + \frac{1}{r} K_\theta \frac{\partial p}{\partial \theta}\right)$$

$$u_\theta = -\frac{1}{\mu}\left(K_\theta \frac{\partial p}{\partial r} + \frac{1}{r} K_t \frac{\partial p}{\partial \theta}\right)$$

Therefore Darcy's law in polar coordinates is represented as:

$$\vec{u} = -\frac{1}{\mu}\bar{K}\nabla_{r,\theta}p = -\frac{1}{\mu}\begin{bmatrix} K_r & K_\theta \\ K_\theta & K_t \end{bmatrix}\begin{bmatrix} \frac{\partial p}{\partial r} \\ \frac{1}{r}\frac{\partial p}{\partial \theta} \end{bmatrix}$$

3.3.3 The Laplace Equation

The Laplace equation, or the Laplacian, is a second order partial differential equation commonly used in many areas of physics, including electromagnetism, astronomy, heat transfer and fluid dynamics (O'Neil, 2003). In this research the Laplace equation is used for material balance purposes. In Cartesian coordinates, the material balance in a source free medium is:

$$\nabla \cdot \vec{u} = 0$$

Using Darcy's law in Cartesian coordinates and assuming the fluid is incompressible, this becomes:

$$\frac{\partial}{\partial x}\left(K_x \frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial p}{\partial y}\right) = 0$$

This is used to determine the pressure at any point throughout the medium providing there is a given set of boundary conditions. If the medium is homogenous, permeability is constant and may be brought outside of the derivative, resulting in:

$$K_x \frac{\partial^2 p}{\partial x^2} + K_y \frac{\partial^2 p}{\partial y^2} = 0$$

Again, solving for a polar coordinate system is the focus of this research. The Laplacian in polar coordinates is:

$$\nabla_{\mathrm{r},\theta} \cdot \vec{u} = 0$$

Using the equations derived previously and again assuming an incompressible fluid this becomes:

$$\nabla_{\mathrm{r},\theta} \cdot \vec{u} = \frac{1}{r}\frac{\partial}{\partial r}(r\, u_r) + \frac{1}{r}\frac{\partial}{\partial \theta}(u_\theta) = 0$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\left(K_r \frac{\partial p}{\partial r} + \frac{1}{r}K_\theta \frac{\partial p}{\partial \theta}\right)\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(K_\theta \frac{\partial p}{\partial r} + \frac{1}{r}K_t \frac{\partial p}{\partial \theta}\right) = 0$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(rK_r \frac{\partial p}{\partial r} + K_\theta \frac{\partial p}{\partial \theta}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(K_\theta \frac{\partial p}{\partial r} + \frac{1}{r}K_t \frac{\partial p}{\partial \theta}\right) = 0$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(rK_r \frac{\partial p}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(K_\theta \frac{\partial p}{\partial \theta}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(K_\theta \frac{\partial p}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(K_t \frac{\partial p}{\partial \theta}\right) = 0$$

This equation is the general Laplacian in polar coordinates for the case when the Cartesian coordinate system aligns with the principle permeability directions.

There are three special cases which result in a simplification of the general Laplacian: isotropic, homogenous or both. In an isotropic medium, $K_r = K_t = K$ and $K_\theta = 0$. The Laplacian reduces to:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(rK\frac{\partial p}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(K\frac{\partial p}{\partial \theta}\right) = 0$$

In the work of Aziz & Settari (1979) the above equation is presented as the general Laplacian. This can only be valid if the permeability matrix in polar coordinates takes the following form:

$$K = \begin{pmatrix} K_r & 0 \\ 0 & K_t \end{pmatrix}$$

This implies $K_\theta = 0$. However, this is only true for isotropic media. For an anisotropic medium K_θ will equal zero only for $\theta = 0, \frac{\pi}{2}, \frac{3\pi}{2}$. These angles occur when the radial direction is parallel to the principle permeability directions (Johansen, 2009). At all angles in between, the general Laplacian presented in the work of Aziz & Settari (1979) will be incorrect for an anisotropic medium.

In a homogenous medium K_r, K_t and K_θ are constants and do not vary with position. The Laplacian reduces to:

$$K_r\frac{\partial^2 p}{\partial r^2} + \frac{K_r}{r}\frac{\partial p}{\partial r} + \frac{K_\theta}{r}\frac{\partial^2 p}{\partial r \partial \theta} + \frac{K_\theta}{r}\frac{\partial^2 p}{\partial \theta \partial r} + \frac{K_t}{r^2}\frac{\partial^2 p}{\partial \theta^2} = 0$$

$$K_r\frac{\partial^2 p}{\partial r^2} + \frac{K_r}{r}\frac{\partial p}{\partial r} + 2\frac{K_\theta}{r}\frac{\partial^2 p}{\partial r \partial \theta} + \frac{K_t}{r^2}\frac{\partial^2 p}{\partial \theta^2} = 0$$

In the simplest case of a homogenous and isotropic medium, both simplifications apply, resulting in:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} + \frac{1}{r^2}\frac{\partial^2 p}{\partial \theta^2} = 0$$

Throughout this chapter the modeling of an isotropic heterogeneous reservoir is discussed. Discussion regarding extending the model to the general case of an anisotropic heterogeneous reservoir is provided within the recommendations for future work in Chapter 6.

3.4 Solving the Laplace Equation

To solve the Laplace equation governing mass balance in the reservoir, several important procedures must first be discussed. The solution to the Laplacian is the pressure distribution throughout the reservoir. Before the solution can be generated, the appropriate modeling procedures must be in place. This includes grid discretization, upscaling, discretization of face velocities, discretization of the Laplacian itself and applying the appropriate boundary conditions. Following this, a system of linear equations may be set in place, leading to the desired solution.

3.4.1 Grid Discretization

To generate a polar grid for a circular reservoir, several inputs are required. First, the number of radial and tangential grid blocks must be determined. The number of radial blocks (N) represents the number of nodes between the wellbore radius and the external reservoir radius. The number of tangential blocks (M) represents the number of sectors to be created within each circle.

Throughout the modeling, specific blocks are referenced by their radial grid number followed by their tangential grid number in the form (i, j). The generalized grid setup is shown in Figure 18. The radial block index, represented by 'i', starts at 1 for the first node after the wellbore and increases with distance away from the wellbore. The tangential block index, represented by 'j', starts at 1 for the first node located half a radial sector away from the conventional x-axis and increase as the angle increases in a counterclockwise direction.

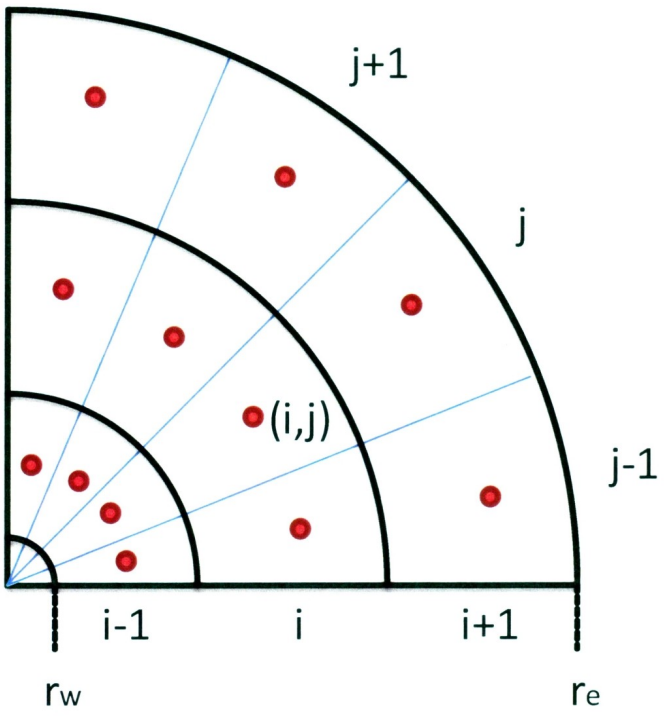


Figure 18 - Grid Discretization

3.4.1.1 Determining Node Distances

In determining the distance between radial nodes, the goal is to set node distances such that the pressure drop between nodes are equal for the special case of a homogenous and isotropic medium. Since pressure drop in a radial geometry is known to be logarithmic, this will result in an increasing number of blocks in the near wellbore region due to the rapid change in pressure. To accommodate this, the following relation is required:

$$r_{i+1} = r_i a = r_w a^{i+1}$$

Note the formulation is set up such that r_1 is the first node, r_0 is the wellbore radius and 'a' is a constant. At the external radius r_e:

$$r_{N+1/2} = r_e = r_w a^{N+1/2}$$

Note the external radius is located at $r_{N+1/2}$ since it is beyond the extent of the final node located at r_N. The expression can be rearranged to solve for the constant:

$$a = \left(\frac{r_e}{r_w}\right)^{1/N+1/2}$$

Therefore the equation to determine the node distances in the radial direction is:

$$r_{i+1} = r_w \left(\frac{r_e}{r_w}\right)^{i+1/N+1/2}$$

Tangentially, the nodes are evenly spaced by determining the change in angle required for the number of tangential blocks selected, and using the general arc length formula ($arc\ length = r \times \Delta\theta$) to place them at their proper locations.

3.4.1.2 Determining Boundary Distances

The boundaries of each grid block must be located between adjacent nodes. The notation $r_{i+1/2}$ represents the boundary distance between nodes i and $i+1$ in the radial direction. This notation is illustrated in Figure 19.

The boundary distances are calculated by determining the flow rate between blocks i and $i+1$:

$$q_{i+1/2} = \frac{2\pi K r_{i+1/2} \Delta z}{\mu} \frac{dp}{dr}$$

In discretized form this becomes:

$$q_{i+1/2} = \frac{2\pi K r_{i+1/2} \Delta z}{\mu} \frac{p_{i+1} - p_i}{r_{i+1} - r_i}$$

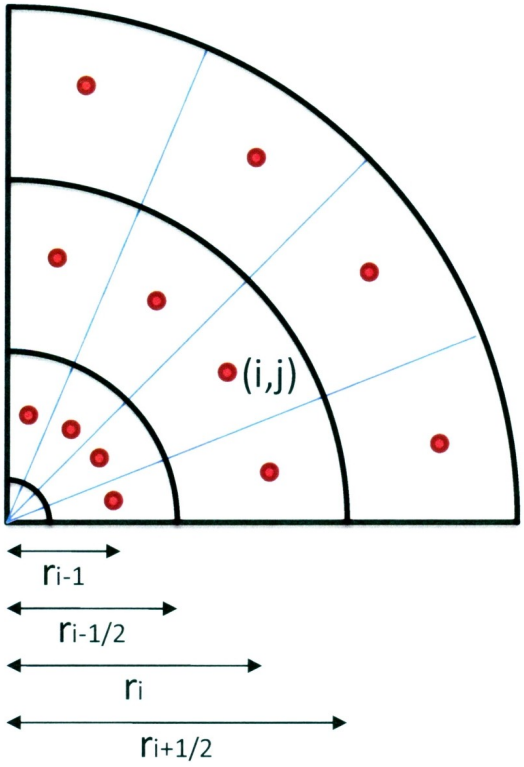


Figure 19 - Radial Node and Boundary Notation

Integrating the expression results in:

$$q_{i+1/2} = \frac{2\pi K \Delta z (p_{i+1} - p_i)}{\mu \, ln\left(\frac{r_{i+1}}{r_i}\right)}$$

Equating both forms of the expression leads to:

$$\frac{2\pi K \Delta z (p_{i+1} - p_i)}{\mu \, ln\left(\frac{r_{i+1}}{r_i}\right)} = \frac{2\pi K r_{i+1/2} \Delta z}{\mu} \frac{p_{i+1} - p_i}{r_{i+1} - r_i}$$

$$r_{i+1/2} = \frac{r_{i+1} - r_i}{ln\left(\frac{r_{i+1}}{r_i}\right)}$$

This formulation is used to determine the radial boundary distances between each node. Tangentially, the division is simply half way between each node.

3.4.2 Upscaling

Upscaling is a processing of averaging permeability when two or more regions need to be merged for the purpose of simplified calculations. In this modeling procedure, flow rate is calculated from block to block, node to node. If adjacent blocks have different permeabilities, as will often occur in a heterogeneous medium, an average or upscaled permeability value must be determined for calculation purposes.

The upscaling procedure will vary depending upon the direction in question. In the radial direction the upscaled radial mobility is a logarithmic average, as derived by Johansen (2008). The general definition of Darcy's Law, neglecting gravity, is:

$$Q = \frac{KA}{\mu}\frac{dp}{dx}$$

Applying this to a homogenous porous cylinder results in:

$$Q = \frac{K2\pi rh}{\mu}\frac{dp}{dr}$$

Evaluating from the external to internal radius results in:

$$\int_{r_w}^{r_e}\frac{dr}{r} = \frac{2\pi Kh}{\mu Q}\int_{p_w}^{p_e} dp$$

$$ln\left(\frac{r_e}{r_w}\right) = \frac{2\pi Kh}{\mu Q}(p_e - p_w)$$

$$Q = \frac{2\pi Kh}{\mu\, ln\left(\frac{r_e}{r_w}\right)}(p_e - p_w)$$

Here we have the general expression of Darcy's Law in a homogenous cylinder. If permeability varies as a function of radius, where $K = K(r)$, we find:

$$\int_{r_w}^{r_e} \frac{dr}{K(r)r} = \frac{2\pi h}{\mu Q} \int_{p_w}^{p_e} dp$$

$$\int_{r_w}^{r_e} \frac{dr}{K(r)r} = \frac{2\pi h}{\mu Q} (p_e - p_w)$$

$$Q = \frac{2\pi h}{\mu \int_{r_w}^{r_e} \frac{dr}{K(r)r}} (p_e - p_w)$$

By definition, the upscaled permeability $\bar{K}$ for radial flow in the cylinder is given by:

$$Q = \frac{2\pi \bar{K} h}{\mu \, ln\left(\frac{r_e}{r_w}\right)} (p_e - p_w)$$

Equating:

$$\frac{2\pi h}{\mu \int_{r_w}^{r_e} \frac{dr}{K(r)r}} (p_e - p_w) = \frac{2\pi \bar{K} h}{\mu \, ln\left(\frac{r_e}{r_w}\right)} (p_e - p_w)$$

$$\frac{1}{\int_{r_w}^{r_e} \frac{dr}{K(r)r}} = \frac{\bar{K}}{ln\left(\frac{r_e}{r_w}\right)}$$

$$\bar{K} = \frac{ln\left(\frac{r_e}{r_w}\right)}{\int_{r_w}^{r_e} \frac{dr}{K(r)r}}$$

If permeability varies in discrete steps, this becomes:

$$\bar{K} = \frac{ln\left(\frac{r_e}{r_w}\right)}{\sum_{i=1}^{N} \frac{1}{K_i} ln\left(\frac{r_i}{r_{i-1}}\right)}$$

The above expression is for applying over a reservoir domain from the external boundary to the internal boundary. Modifying this to evaluate over two internal intervals (adjacent grid blocks), and applying strictly for mobility, this becomes:

$$\lambda^r_{i+1/2} = \frac{ln\left(\frac{r_{i+1}}{r_i}\right)}{\frac{1}{\lambda^r_i} ln\left(\frac{r_{i+1/2}}{r_i}\right) + \frac{1}{\lambda^r_{1+1}} ln\left(\frac{r_{i+1}}{r_{i+1/2}}\right)}$$

Note that mobility (λ) is permeability divided by viscosity, or $\lambda = K/\mu$. Since viscosity is assumed constant, the upscaling procedure is performed for mobility as opposed to permeability. The results are indifferent. Also note the superscript on λ^r represents the radial direction, while the superscript on λ^t represents the tangential direction.

The upscaled tangential mobility is equivalent to serial flow through parallel layers. This is a harmonic mean in which the upscaled formula becomes:

$$\frac{1}{\lambda^t_{j+1/2}} = \frac{1}{L}\int_0^L \frac{dx}{\lambda^t(x)} = \frac{1}{L}\sum \frac{L_j}{\lambda^t{}_j}$$

$$\lambda^t_{j+1/2} = \frac{2\lambda^t{}_j \lambda^t{}_{j+1}}{\lambda^t{}_j + \lambda^t{}_{j+1}}$$

3.4.3 Discretization of Face Velocities

The face velocities must be discretized across all grid block faces for the purposes of the streamline calculations to follow. Figure 20 shows a schematic of the face velocity notation for a given grid block (i, j).

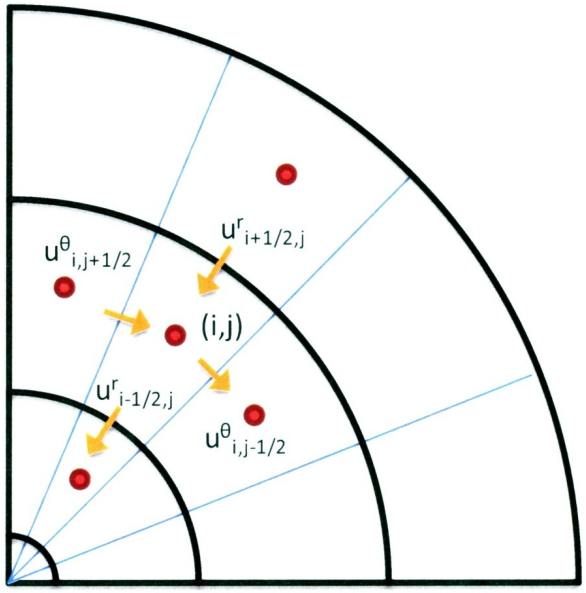


Figure 20 - Discretization of Face Velocities

The figure indicates the positive directions for the radial and tangential velocities. Radially, velocity is considered to be positive inwards. Tangential, velocity is considered positive in the clockwise direction. Note that this is opposite to the chosen convention for an increase in angle, which is positive in the counterclockwise direction. Also note that the velocities mentioned here are not true velocities; they are Darcy velocities or volumetric fluxes. Later in the streamline modeling calculations the volumetric fluxes must be converted to real velocities through the use of $V = u/\emptyset$ for time of flight calculations, but this is not required at this stage in the modeling procedure.

The general flux formulae in polar coordinates, as derived previously, are:

$$u_r = -\frac{1}{\mu}\left(K_r \frac{\partial p}{\partial r} + \frac{1}{r} K_\theta \frac{\partial p}{\partial \theta}\right)$$

$$u_\theta = -\frac{1}{\mu}\left(K_\theta \frac{\partial p}{\partial r} + \frac{1}{r} K_t \frac{\partial p}{\partial \theta}\right)$$

Therefore in discretized form, and noting $K_\theta = 0$ for the isotropic case, the flux formulations become:

$$u^r_{i+1/2,j} = \lambda^r_{i+1/2,j} \frac{p_{i+1,j} - p_{i,j}}{r_{i+1} - r_i}$$

$$u^r_{i-1/2,j} = \lambda^r_{i-1/2,j} \frac{p_{i,j} - p_{i-1,j}}{r_i - r_{i-1}}$$

$$u^\theta_{i,j+1/2} = \frac{1}{r_i} \lambda^t_{i,j+1/2} \frac{p_{i,j+1} - p_{i,j}}{\Delta\theta}$$

$$u^\theta_{i,j-1/2} = \frac{1}{r_i} \lambda^t_{i,j-1/2} \frac{p_{i,j} - p_{i,j-1}}{\Delta\theta}$$

Note that the negative signs have been dropped to be consistent with the directions shown in Figure 20.

3.4.4 Discretization of the Laplacian

Recall the Laplace equation for a heterogeneous isotropic reservoir in polar coordinates is:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(rK\frac{\partial p}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(K\frac{\partial p}{\partial \theta}\right) = 0$$

Reformulating this for a discrete mathematical model, or finite difference, results in:

$$\frac{1}{r_i\left(r_{i+1/2} - r_{i-1/2}\right)}\left[r_{i+1/2}u^r_{i+1/2,j} - r_{i-1/2}u^r_{i-1/2,j}\right] + \frac{1}{r_i}\frac{1}{\Delta\theta}\left[u^\theta_{i,j+1/2} - u^\theta_{i,j-1/2}\right] = 0$$

To further evaluate this formulation the formulae for the Darcy velocities are inserted into the Laplacian, resulting in:

$$\frac{1}{r_i(r_{i+1/2}-r_{i-1/2})}\left[r_{i+1/2}\left(\lambda^r_{i+1/2,j}\frac{p_{i+1,j}-p_{i,j}}{r_{i+1}-r_i}\right)-r_{i-1/2}\left(\lambda^r_{i-1/2,j}\frac{p_{i,j}-p_{i-1,j}}{r_i-r_{i-1}}\right)\right]$$

$$+\frac{1}{r_i}\frac{1}{\Delta\theta}\left[\left(\frac{1}{r_i}\lambda^t_{i,j+1/2}\frac{p_{i,j+1}-p_{i,j}}{\Delta\theta}\right)-\left(\frac{1}{r_i}\lambda^t_{i,j-1/2}\frac{p_{i,j}-p_{i,j-1}}{\Delta\theta}\right)\right]=0$$

Rewriting this equation such that it becomes a system of solvable linear equations results in:

$$\left(\frac{r_{i+1/2}\lambda^r_{i+1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_{i+1}-r_i)}\right)(p_{i+1,j}-p_{i,j})$$

$$-\left(\frac{r_{i-1/2}\lambda^r_{i-1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_i-r_{i-1})}\right)(p_{i,j}-p_{i-1,j})$$

$$+\left(\frac{\lambda^t_{i,j+1/2}}{r_i{}^2\Delta\theta^2}\right)(p_{i,j+1}-p_{i,j})-\left(\frac{\lambda^t_{i,j-1/2}}{r_i{}^2\Delta\theta^2}\right)(p_{i,j}-p_{i,j-1})=0$$

Separation of the pressure terms, grouping variables and further reduction leads to:

$$\left(\frac{r_{i+1/2}\lambda^r_{i+1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_{i+1}-r_i)}\right)p_{i+1,j}+\left(\frac{r_{i-1/2}\lambda^r_{i-1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_i-r_{i-1})}\right)p_{i-1,j}$$

$$+\left(\frac{\lambda^t_{i,j+1/2}}{r_i{}^2\Delta\theta^2}\right)p_{i,j+1}+\left(\frac{\lambda^t_{i,j-1/2}}{r_i{}^2\Delta\theta^2}\right)p_{i,j-1}$$

$$-\left[\left(\frac{r_{i+1/2}\lambda^r_{i+1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_{i+1}-r_i)}\right)+\left(\frac{r_{i-1/2}\lambda^r_{i-1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_i-r_{i-1})}\right)\right.$$

$$\left.+\left(\frac{\lambda^t_{i,j+1/2}+\lambda^t_{i,j-1/2}}{r_i{}^2\Delta\theta^2}\right)\right]p_{i,j}=0$$

This is the general form of the discretized Laplacian for an isotropic heterogeneous reservoir. Rearranging in the following format:

$$a(i,j)p_{i,j}+b(i,j)p_{i,j+1}+c(i,j)p_{i,j-1}+d(i,j)p_{i-1,j}+e(i,j)p_{i+1,j}=0$$

The general expression for the coefficients becomes:

$$a(i,j) = -\left[\left(\frac{r_{i+1/2}\lambda^r_{i+1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_{i+1}-r_i)}\right)+\left(\frac{r_{i-1/2}\lambda^r_{i-1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_i-r_{i-1})}\right)+\left(\frac{\lambda^t_{i,j+1/2}+\lambda^t_{i,j-1/2}}{r_i{}^2\Delta\theta^2}\right)\right]$$

$$b(i,j) = \left(\frac{\lambda^t_{i,j+1/2}}{r_i{}^2\Delta\theta^2}\right)$$

$$c(i,j) = \left(\frac{\lambda^t_{i,j-1/2}}{r_i{}^2\Delta\theta^2}\right)$$

$$d(i,j) = \left(\frac{r_{i-1/2}\lambda^r_{i-1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_i-r_{i-1})}\right)$$

$$e(i,j) = \left(\frac{r_{i+1/2}\lambda^r_{i+1/2,j}}{r_i(r_{i+1/2}-r_{i-1/2})(r_{i+1}-r_i)}\right)$$

The general equation must be modified to account for the known boundary conditions in order to make a solvable system of equations to determine the pressure at each node. There are five pressure terms for each line within the equation. Therefore the solution to one block depends upon the conditions of the block itself plus its four surrounding blocks.

3.4.5 Applying Laplace Boundary Conditions

To determine the solution to the discretized Laplacian several boundary conditions must be imposed. The boundary conditions involve areas of known pressure and allow the solution to be generated. There are two main boundary conditions; one at the wellbore

and one at the external reservoir radius. First, the pressure at the wellbore, P_{wf}, is a known constant. This condition is summarized as: $P_0 = P_{wf}$ at $r_w = r_0$. Second, the pressure at the external radius, P_e, is a known constant. This condition is summarized as: $P_{N+1} = P_e$ at $r_e = r_{N+1}$.

Another boundary condition involving grid crossover is also important. For a total of M tangential grid blocks, once the tangential counter j reaches $j = M$, $j + 1$ becomes equal to $j = 1$. Likewise, when the tangential counter reaches $j = 1$, $j - 1$ becomes equal to $j = M$. These conditions are required for proper grid block referencing in the polar grid.

In the following subsection the application of the boundary conditions to the Laplace equation are discussed.

3.4.5.1 Wellbore Ring

At the wellbore, the internal boundary condition must be applied. Therefore nodes located at $i = 1$ and $j = 1, \dots, M$ must have the following boundary conditions imposed:

$$p_{i-1,j} = P_{wf} \qquad r_{i-1/2} = r_w \qquad r_{i-1} = r_w$$

Note that the internal ring $(r_{i-1/2})$ and the internal node (r_{i-1}) are both equal to the wellbore radius (r_w). This is required for modeling purposes and does not represent an inaccuracy in the solution.

Since $p_{i-1,j}$ is known and equal to p_{wf}, the known quantities can be brought to the right hand side of the general Laplace equation, as shown below:

$$a(1,j)p_{i,j} + b(1,j)p_{i,j+1} + c(1,j)p_{i,j-1} + e(1,j)p_{i+1,j} = -d(1,j)P_{wf}$$

3.4.5.2 External Ring

At the external radius, the external boundary condition must be applied. Therefore nodes located at $i = N$ and $j = 1, \dots, M$ must have the following boundary conditions imposed:

$$p_{i+1,j} = P_e \qquad r_{i+1/2} = r_e \qquad r_{i+1,j} = r_e$$

Note that the external ring $(r_{N+1/2})$ and the external node (r_{N+1}) are both equal to the external radius (r_e). As with the internal boundary condition, this is required for modeling purposes and does not represent an inaccuracy in the solution.

Since $p_{i+1,j}$ is known and equal to p_e, the known quantities can be brought to the right hand side of the general Laplace equation, taking the following form:

$$a(N,j)p_{i,j} + b(N,j)p_{i,j+1} + c(N,j)p_{i,j-1} + d(N,j)p_{i-1,j} = -e(N,j)P_e$$

3.4.6 Solving a System of Linear Equations

The Laplacian in the format expressed in the previous section forms a system of linear equations. There are a total of $N \times M$ unknowns and $N \times M$ equations, resulting in a solvable system. For solution the system of equations is written in matrix form. For simplicity and memory saving purposes the system matrix is desired to have a minimum bandwidth, thereby keeping the system matrix size to a minimum. The system matrix is organized as follows:

$$\vec{P} = \left[\underbrace{p_{1,1}, p_{1,2}, \dots, p_{1,M}}_{i=1}, \underbrace{p_{2,1}, p_{2,2}, \dots, p_{2,M}}_{i=2}, \dots, \underbrace{p_{N,1}, p_{N,2}, \dots, p_{N,M}}_{i=N}\right]$$

Here the pressure solution is a column matrix in which the node pressure values for all elements of an internal ring are inputted first prior to moving to the next ring. Recall that from the previous section the generalized equations are:

For $i = 1$:

$$a(1,j)p_{i,j} + b(1,j)p_{i,j+1} + c(1,j)p_{i,j-1} + e(1,j)p_{i+1,j} = -d(1,j)P_{wf}$$

For $i = 2, \dots, N-1$:

$$a(i,j)p_{i,j} + b(i,j)p_{i,j+1} + c(i,j)p_{i,j-1} + d(i,j)p_{i-1,j} + e(i,j)p_{i+1,j} = 0$$

For $i = N$:

$$a(N,j)p_{i,j} + b(N,j)p_{i,j+1} + c(N,j)p_{i,j-1} + d(N,j)p_{i-1,j} = -e(N,j)P_e$$

For clarity, a simple case of $N = 4$ and $M = 5$ is evaluated. The system equations, placed in proper matrix format, will take the following format:

$$\begin{pmatrix}
a_{1,1} & b_{1,1} & 0 & 0 & c_{1,1} & e_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_{1.2} & a_{1.2} & b_{1.2} & 0 & 0 & 0 & e_{1.2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & c_{1,3} & a_{1,3} & b_{1,3} & 0 & 0 & 0 & e_{1,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c_{1.4} & a_{1.4} & b_{1.4} & 0 & 0 & 0 & e_{1.4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b_{1.5} & 0 & 0 & c_{1.5} & a_{1.5} & 0 & 0 & 0 & 0 & e_{1.5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_{2.1} & 0 & 0 & 0 & 0 & a_{2.1} & b_{2.1} & 0 & 0 & c_{2.1} & e_{2.1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & d_{2,2} & 0 & 0 & 0 & c_{2,2} & a_{2,2} & b_{2,2} & 0 & 0 & 0 & e_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & d_{2,3} & 0 & 0 & 0 & c_{2,3} & a_{2,3} & b_{2,3} & 0 & 0 & 0 & e_{2,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & d_{2.4} & 0 & 0 & 0 & c_{2.4} & a_{2.4} & b_{2.4} & 0 & 0 & 0 & e_{2.4} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & d_{2.5} & b_{2.5} & 0 & 0 & c_{2.5} & a_{2.5} & 0 & 0 & 0 & 0 & e_{2.5} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & d_{3,1} & 0 & 0 & 0 & 0 & a_{3,1} & b_{3,1} & 0 & 0 & c_{3,1} & e_{3,1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & d_{3,2} & 0 & 0 & 0 & c_{3,2} & a_{3,2} & b_{3,2} & 0 & 0 & 0 & e_{3,2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{3,3} & 0 & 0 & 0 & c_{3,3} & a_{3,3} & b_{3,3} & 0 & 0 & 0 & e_{3,3} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{3.4} & 0 & 0 & 0 & c_{3.4} & a_{3.4} & b_{3.4} & 0 & 0 & 0 & e_{3.4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{3,5} & b_{3,5} & 0 & 0 & c_{3,5} & a_{3,5} & 0 & 0 & 0 & 0 & e_{3,5} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{4,1} & 0 & 0 & 0 & 0 & a_{4,1} & b_{4,1} & 0 & 0 & c_{4,1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{4,2} & 0 & 0 & 0 & c_{4,2} & a_{4,2} & b_{4,2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{4,3} & 0 & 0 & 0 & c_{4,3} & a_{4,3} & b_{4,3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{4,4} & 0 & 0 & 0 & c_{4,4} & a_{4,4} & b_{4,4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{4,5} & b_{4,5} & 0 & 0 & c_{4,5} & a_{4,5}
\end{pmatrix}
\begin{pmatrix}
p_{1,1} \\ p_{1,2} \\ p_{1,3} \\ p_{1,4} \\ p_{1,5} \\ p_{2,1} \\ p_{2,2} \\ p_{2,3} \\ p_{2,4} \\ p_{2,5} \\ p_{3,1} \\ p_{3,2} \\ p_{3,3} \\ p_{3,4} \\ p_{3,5} \\ p_{4,1} \\ p_{4,2} \\ p_{4,3} \\ p_{4,4} \\ p_{4,5}
\end{pmatrix} =
\begin{pmatrix}
-d_{1,1}\, p_{wf} \\ -d_{1.2}\, p_{wf} \\ -d_{1,3}\, p_{wf} \\ -d_{1.4}\, p_{wf} \\ -d_{1.5}\, p_{wf} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -e_{4,1}\, p_e \\ -e_{4,2}\, p_e \\ -e_{4,3}\, p_e \\ -e_{4,4}\, p_e \\ -e_{4,5}\, p_e
\end{pmatrix}$$

Note the location of the coefficients in the coefficient matrix. The locations of b and c vary as a result of being multiplied by $p_{i,j+1}$ and $p_{i,j-1}$, respectively. The grid boundary condition involving the crossover at $j = M$ and $j = 1$ are responsible for this movement.

The matrix equation can be expressed as:

$$[A][P] = [B]$$

This may be solved using the inverse of the system matrix or other decomposition techniques. The inverse matrix solution is:

$$[P] = [A]^{-1}[B]$$

In the code generated for this model, provided in Appendix I, the matrix inverse technique is utilized. Matrix inversion for large matrices requires significant memory capacity. However, the system matrix requiring inversion is sparse, mostly populated with zeros. Taking advantage of this, the system matrix is made with the sparse command in MATLAB. This process coverts the full matrix to a sparse matrix by squeezing out any zero elements. In doing so, much less information is stored. This allows the matrix inversion to be carries out much quicker and more efficient, and also allows for the processing of much finer grid resolutions.

3.5 Reservoir Pressure Distribution: General Case Study Results

The solution to the Laplacian for pressure distribution through the reservoir medium is a function of the various input parameters, the fundamental equation formulation and the boundary conditions imposed. The equations derived in the previous section are for a heterogeneous isotropic reservoir. Therefore, while permeability may vary with location $\left(k_{i,j} \neq k_{i+1,j+1}\right)$, it cannot vary in its principle directions $\left(k_{i,j}^{r} = k_{i,j}^{t}\right)$.

The following sections show the results of several generalized case studies. Three cases are discussed; a homogenous reservoir, a reservoir with a large heterogeneity of low permeability and a reservoir with a large sector of high permeability. The cases are not meant to be realistic scenarios with respect to the heterogeneity size and distribution. They are provided for verification purposes to show the effectiveness and success of the calculation procedure. Realistic scenarios are discussed in Chapter 4.

3.5.1 Case A: Homogenous Reservoir

A homogenous reservoir is a special case of a heterogeneous reservoir where the permeability throughout the reservoir is constant. The details of the specific parameters used in this case are shown in Table 1.

Figure 21 provides a three dimensional plot displaying the solution to the Laplacian for the homogenous case. The x-axis and y-axis represent the spatial location throughout the reservoir, while the z-axis represents the corresponding pressure.

Table 1 – Case A: Parameter Details

Case A	Homogenous reservoir ran to demonstrate success of pressure distribution model			
Factor	**Symbol**	**Name**	**Actual Units**	**Value**
A	r_w	Wellbore Radius	m	0.05
B	r_e	External Radius	m	50
C	N	Radial Blocks		50
D	M	Tangential Blocks		120
E	p_{wf}	Wellbore Pressure	bar	280
F	p_e	External Pressure	bar	300
G	k	Bulk Permeability	Darcy	1
H	h	Reservoir Thickness	m	5
I	μ	Oil Viscosity	cP	1

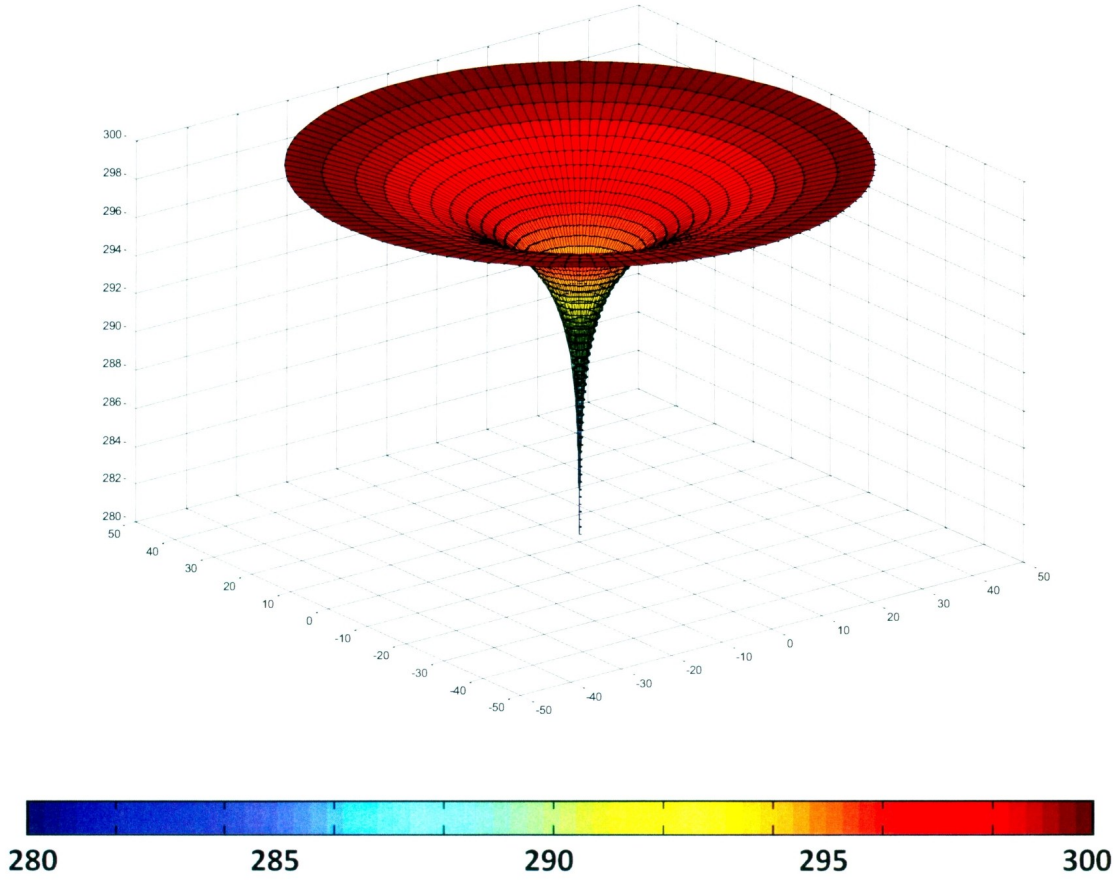


Figure 21 - Case A: 3D Pressure Distribution

As can be seen from the figure, the pressure distribution is symmetric throughout the reservoir with no disruptions as a result of permeability heterogeneities. The color represents the pressure in accordance with the color bar.

From Figure 21 it is clearly noticeable that the pressure throughout the reservoir drops in a logarithmic fashion as it approaches the wellbore. Therefore the greatest pressure drops occur in the near wellbore region. This is as a result of substantial flow accumulation in a diminishing area. Figure 22 provides a plan view of the reservoirs pressure distribution.

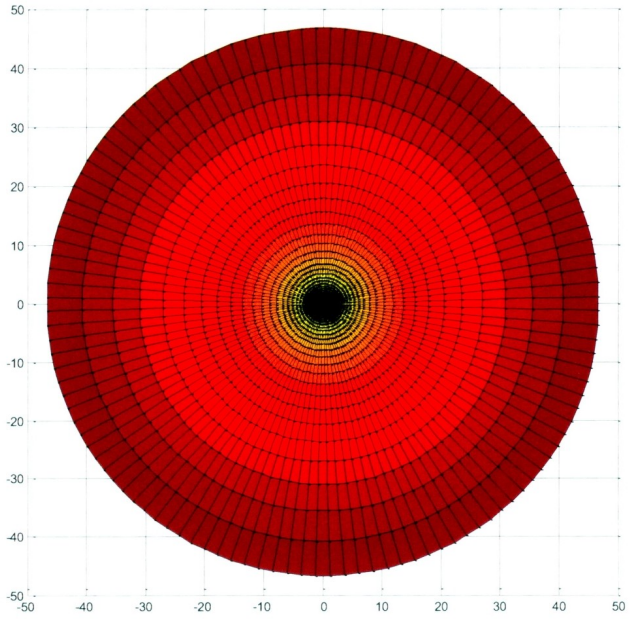
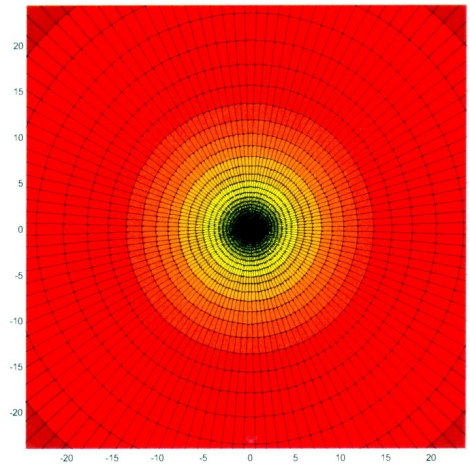
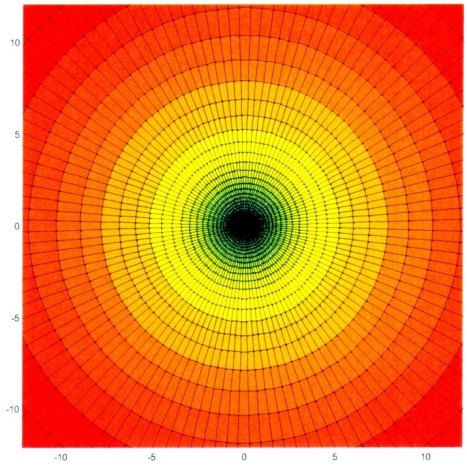


Figure 22 - Case A: Plan View of Pressure Distribution

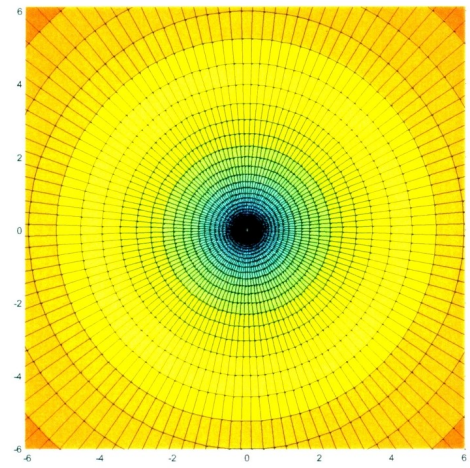
Due to the scale of the figure the majority of the pressure drop is unrecognizable since the near wellbore area is small in comparison to the size of the reservoir. Figure 23 provides a progressive zoom-in of the near wellbore area to emphasize the pressure loss in this vicinity.



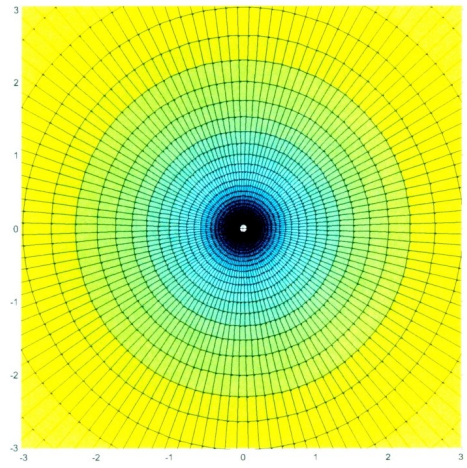
Zoom (2x)



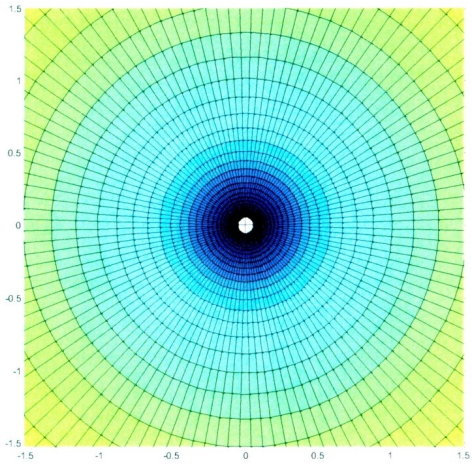
Zoom (3x)



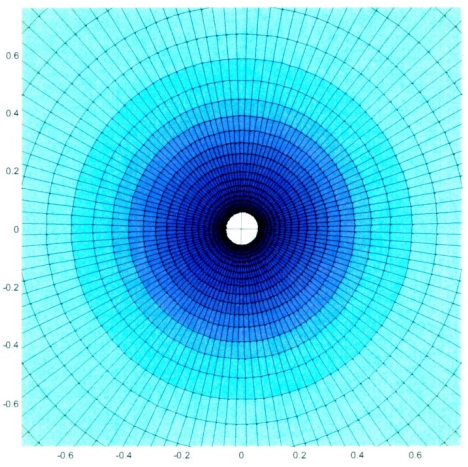
Zoom (4x)



Zoom (5x)



Zoom (6x)



Zoom (7x)

Figure 23 - Case A: Pressure Loss in Near Wellbore Region (Zoom 2X to 7X)

3.5.2 Case B: Heterogeneity with Low Permeability

In this case a large heterogeneity of low permeability is placed in the second quadrant of the reservoir in relative close proximity to the wellbore. The details of the specific parameters used in this case are provided in Table 2.

Table 2 – Case B: Parameter Details

Case B	Block of low permeability to demonstrate the effect on overall pressure distribution			
Factor	**Symbol**	**Name**	**Actual Units**	**Value**
A	r_w	Wellbore Radius	m	0.05
B	r_e	External Radius	m	50
C	N	Radial Blocks		50
D	M	Tangential Blocks		120
E	p_{wf}	Wellbore Pressure	bar	280
F	p_e	External Pressure	bar	300
G	k	Bulk Permeability	Darcy	1
H	h	Reservoir Thickness	m	5
I	μ	Oil Viscosity	cP	1
J		Radial Blocks with Low Permeability		25-41
K		Tangential Blocks with Low Permeability		45-60
L		Block Permeability	Darcy	0.05

Figure 24 provides a plan view of the permeability distribution in the reservoir. The block of low permeability is clearly visible in blue. Figure 25 provides a close up of this region to better identify the location with respect to the wellbore.

The effect of the heterogeneity on the pressure distribution in the reservoir can be seen in Figure 26 and Figure 27.

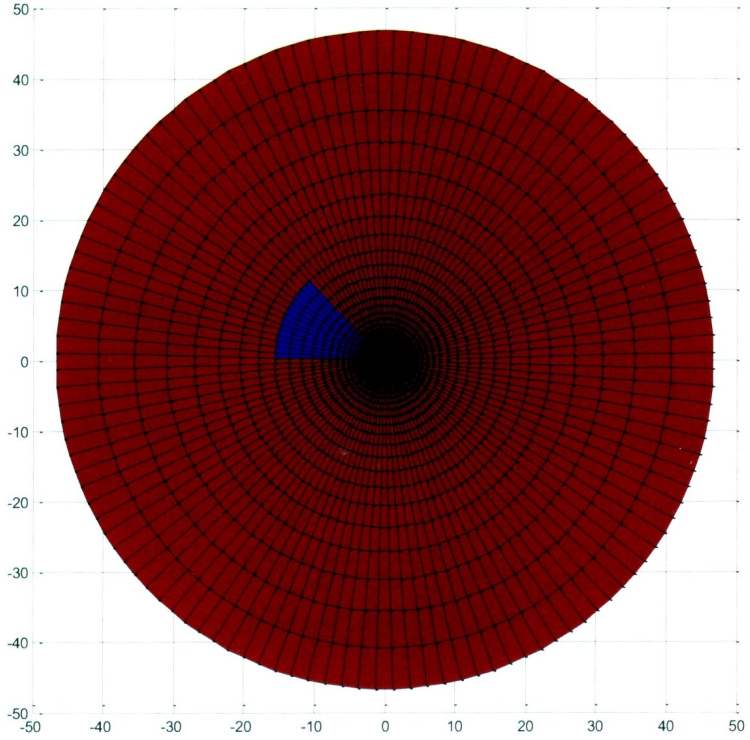


Figure 24 - Case B: Permeability Distribution

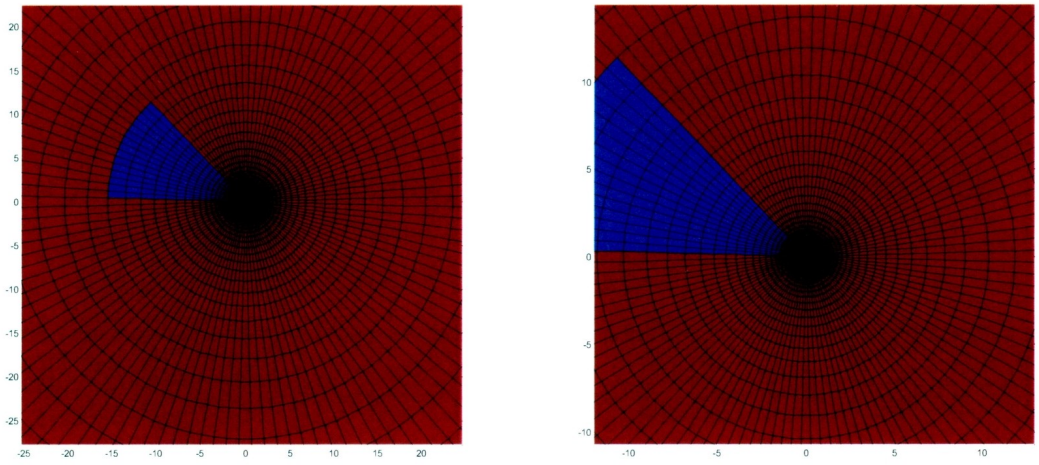


Figure 25 - Case B: Close-up of Permeability Distribution

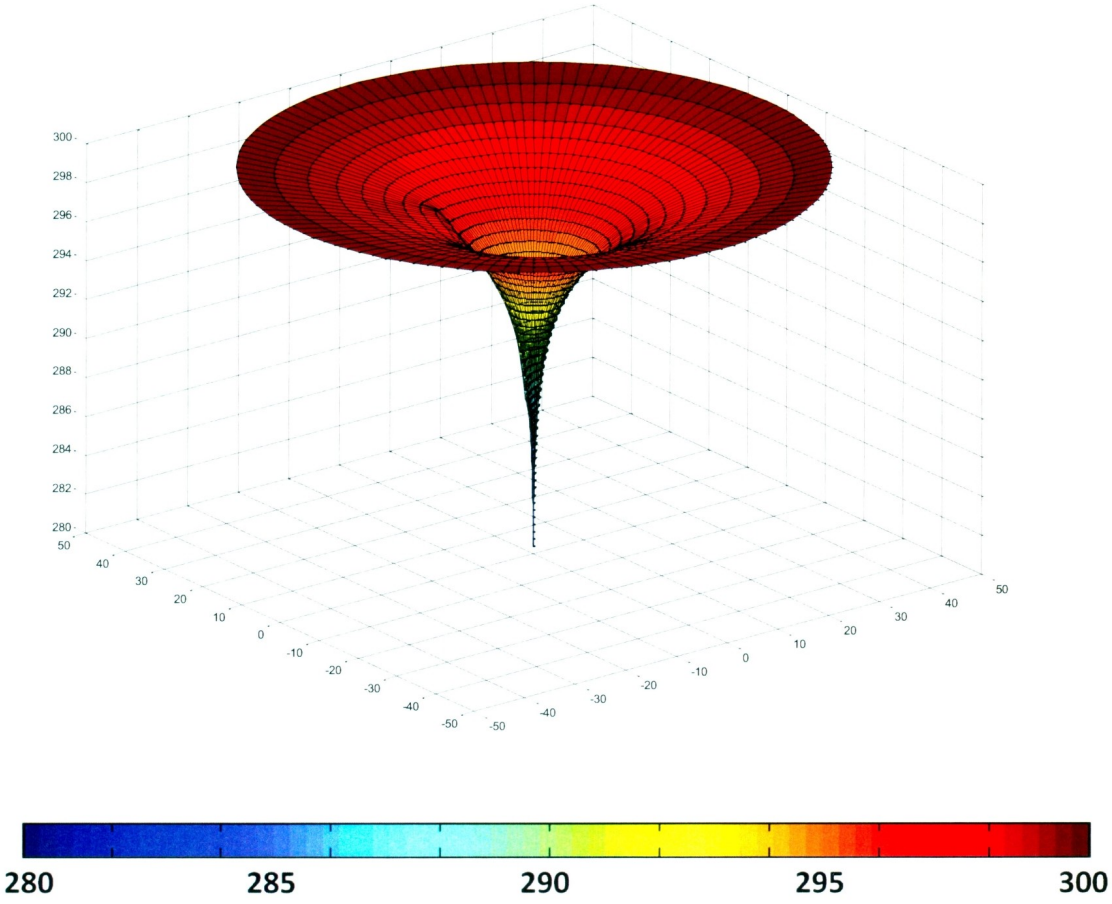


Figure 26 - Case B: 3D Pressure Distribution

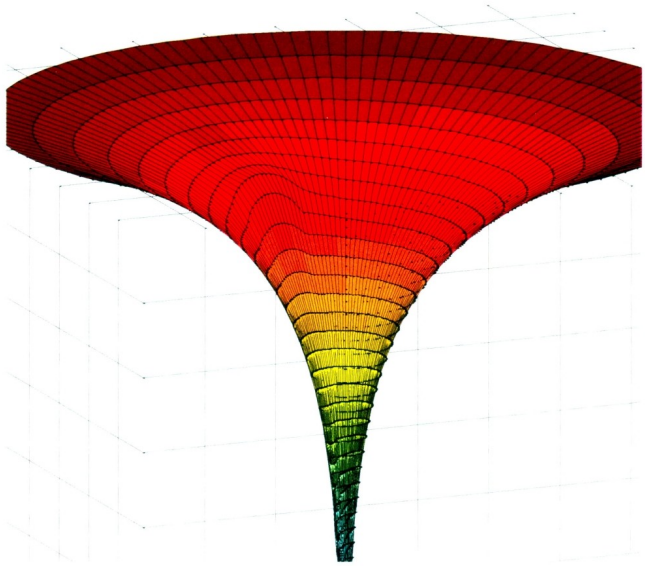


Figure 27 - Case B: Close-up of Most Significantly Affected Area

As can be seen in the figures, the heterogeneity has a significant effect on the pressure distribution around it. The lower permeability value leads to a larger pressure drop within the heterogeneity. This must also affect the surrounding pressure distribution, since pressure is a continuous variable. Therefore, the pressure distribution molds itself around the low permeability area to account for the large pressure drop, resulting in smaller pressure drops surrounding the region.

3.5.3 Case C: Fractured Sector with High Permeability

In the final preliminary case a large sector of high permeability is placed in the second quadrant of the reservoir. The case is somewhat similar to that of a fracture, but does not represent a realistic fracture due to the size of the region chosen. The details of the specific parameters used in this case are provided in Table 3.

Table 3 – Case C: Parameter Details

Case C	Sector of high permeability to demonstrate the effect on overall pressure distribution			
Factor	**Symbol**	**Name**	**Actual Units**	**Value**
A	r_w	Wellbore Radius	m	0.05
B	r_e	External Radius	m	50
C	N	Radial Blocks		50
D	M	Tangential Blocks		120
E	p_{wf}	Wellbore Pressure	bar	280
F	p_e	External Pressure	bar	300
G	k	Bulk Permeability	Darcy	1
H	h	Reservoir Thickness	m	5
I	μ	Oil Viscosity	cP	1
J		Radial Blocks with Low Permeability		all
K		Tangential Blocks with Low Permeability		45-60
L		Block Permeability	Darcy	10

Figure 28 provides a plan view of the permeability distribution in the reservoir. The sector of high permeability is clearly visible in red.

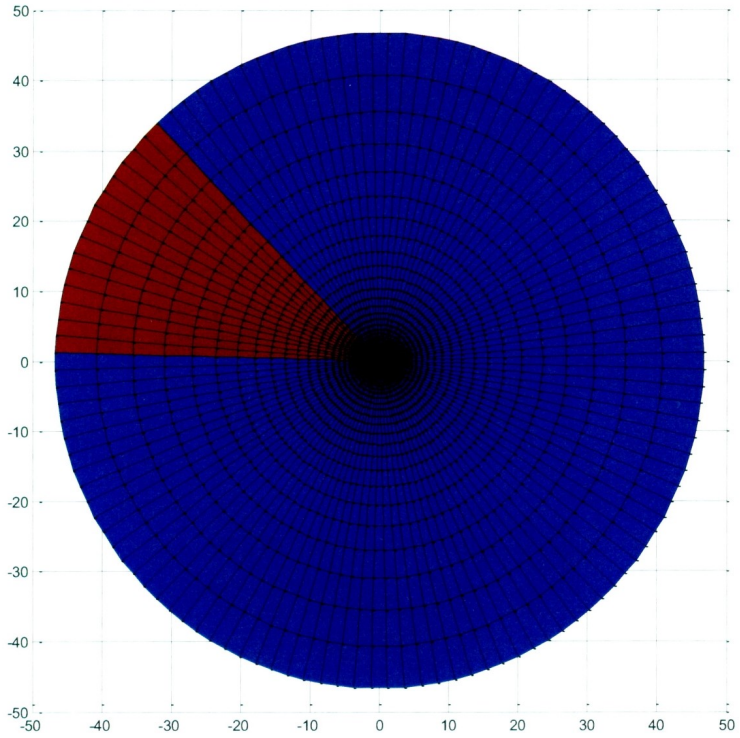


Figure 28 - Case C: Permeability Distribution

The effect of the high permeability sector on the pressure distribution in the reservoir can be seen in Figure 29 and Figure 30.

The region of high permeability is clearly evident in the figures, having a significant effect on the pressure distribution throughout the reservoir. The high permeability within the sector leads to low pressure drops within the sector. To accommodate to the lower pressure in the vicinity, significant pressure drops occur around the sector. The large differential pressure around the sector will lead to an influx of fluids in the region. Only

the pressure can be seen thus far. To understand the nature of the flow profiles, streamline models must be invoked.

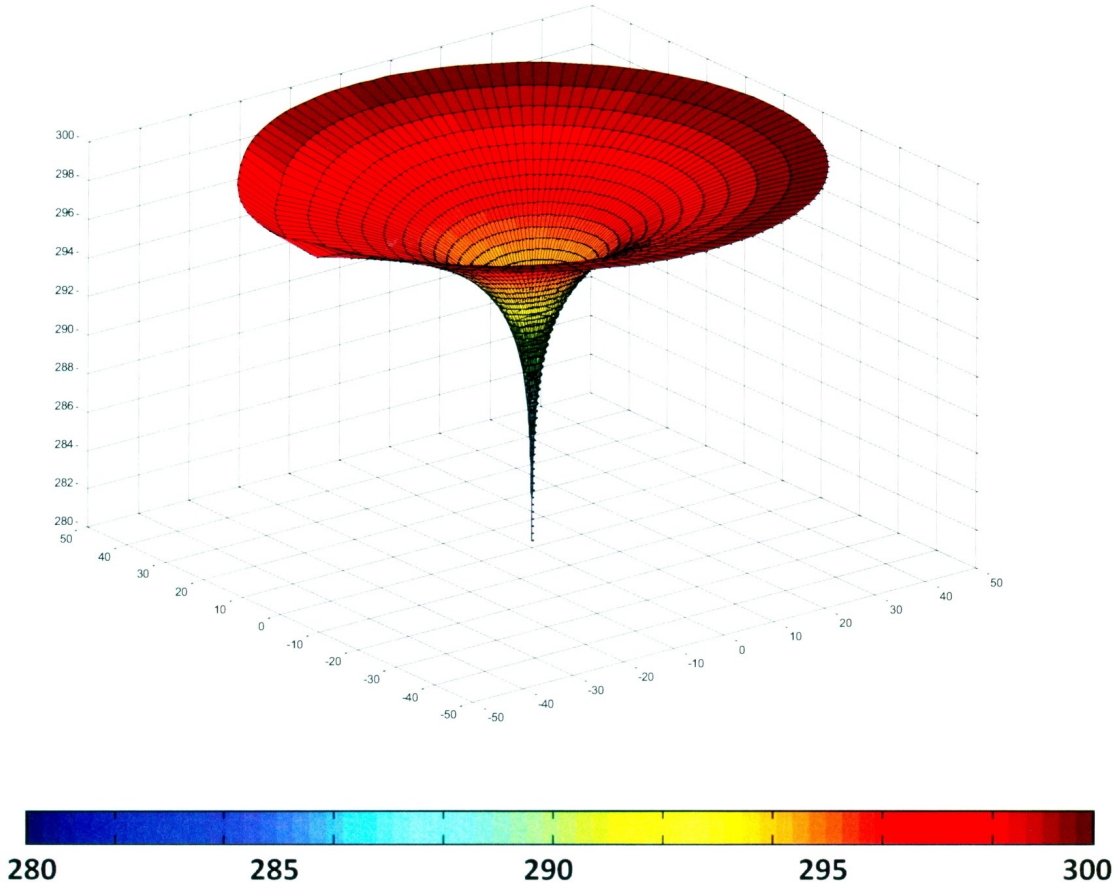


Figure 29 - Case C: 3D Pressure Distribution

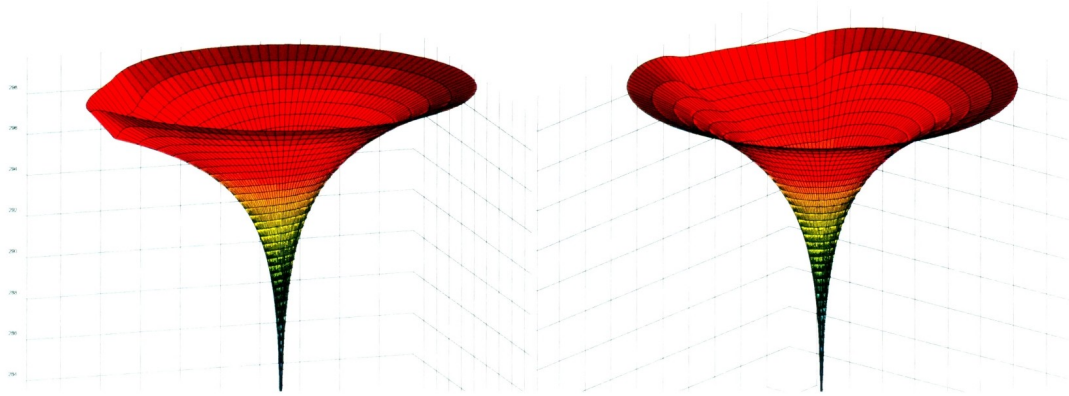


Figure 30 - Case C: Different Viewpoints of the Pressure Distribution

3.6 Streamline Modeling in Radial Geometry

The radial finite difference grid along with the Laplace solution provides a view of the pressure distribution throughout a cylindrical reservoir. Beyond the pressure distribution, the goal of this research is to understand the nature of flow throughout the reservoir, particularly in the near wellbore region. To accomplish this, streamline models must be invoked. The basic definition of streamlines is discussed in Chapters 1 and 2, as well as Section 3.1 of this chapter. The remainder of this section describes the methodology for creating streamlines once the pressure distribution is known.

3.6.1 Pollock's Particle Tracking Methodology

As discussed in Chapter 2, the particle tracking method utilized in the modern day streamline approach was first published by David Pollock (1988). He presented a simple concept to track the movement of a particle through a porous medium based upon the known velocity field. If the velocity is known at all points throughout the medium, the movement of a particle can be determined.

To start, a finite difference grid is created in which the pressure solution at all grid nodes must be known. Once the pressure at each node is determined, the velocities along each grid boundary can be determined using the adjacent node pressures. This enables the velocities along all grid block faces to be determined.

For a fully analytical model, the velocity at all points in the medium must be known. However, for a finite difference grid the velocities across faces are known, while the velocities at points within the grid blocks must be determined through some form of interpolation. This creates a semi-analytical procedure for streamline modeling.

Pollock's initial paper is derived for a Cartesian coordinate grid. It is revolutionary in the area of streamline modeling due to the introduction of one simple concept; time of flight. Figure 31 provides the Cartesian grid setup necessary to understand the concept.

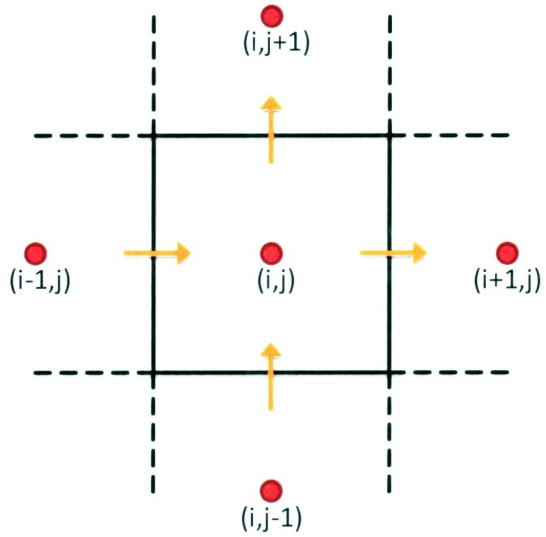


Figure 31 - Pollock Cartesian Methodology

In the figure red dots represent grid block nodes, while the black lines represent boundaries between grid blocks. Using the differential pressure between nodes the face velocities can be determined, as shown by the orange arrows in the figure. The remainder of the velocity field is determined by linear interpolation between boundaries.

The next step in the process involves the introduction of an entrance point. This is a point on a grid block face in which a particle first enters the block. Knowing the location of the point, two travel times can be determined; one for the time to reach the vertical boundary (t_2) and one for the time to reach the horizontal boundary (t_1). The distance is a function of the location of the entrance point, while the time will also be related to the velocity in each direction. Figure 32 highlights the two possible scenarios.

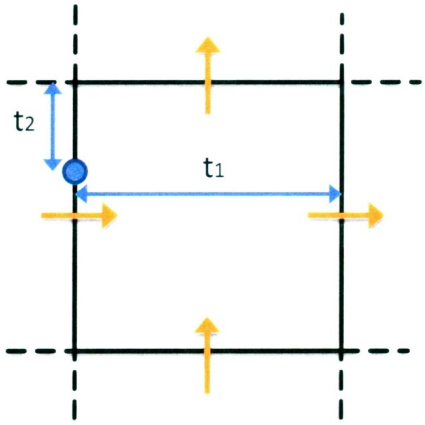


Figure 32 - Pollock Travel Options

The time of flight concept refers to the time in which a particle is within a block. This time is the lesser of the two times to reach the two grid block boundaries. Therefore, by determining which grid boundary is reached first, the calculated time may be placed into the equation for the opposite direction to determine the precise exit location based upon the velocity field. In Figure 32, an example of a particle entering the west face with t_2 as the minimum time is provided. The location of the exit point would be determined by the distance travelled horizontally in the time allotted. The entrance and exit points are shown in Figure 33.

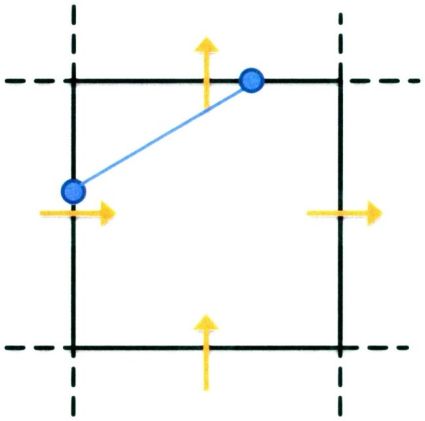


Figure 33 - Pollock Exit Point

The calculation process is continued for the remaining grid blocks, as the exit point for one block is always the entrance point to another. The continuation of this process from the initial point to the final point creates a streamline, showing the path of a particle throughout the porous medium.

This general explanation is provided to aid in the understanding of the process to be further developed throughout the remainder of this section. Several of the intricacies in determining the exit location for various directional velocity combinations have yet to be discussed. As well, the remainder of this discussion is focused solely on polar geometry, in which several differences from the Pollock method exist.

3.6.2 Velocity Formulation in Radial Geometry

As shown in Section 3.4.3, the discretized face velocities in polar coordinates are:

$$u^r_{i+1/2,j} = \lambda^r_{i+1/2,j} \frac{p_{i+1,j} - p_{i,j}}{r_{i+1} - r_i}$$

$$u^r_{i-1/2,j} = \lambda^r_{i-1/2,j} \frac{p_{i,j} - p_{i-1,j}}{r_i - r_{i-1}}$$

$$u^\theta_{i,j+1/2} = \frac{1}{r_i} \lambda^t_{i,j+1/2} \frac{p_{i,j+1} - p_{i,j}}{\Delta\theta}$$

$$u^\theta_{i,j-1/2} = \frac{1}{r_i} \lambda^t_{i,j-1/2} \frac{p_{i,j} - p_{i,j-1}}{\Delta\theta}$$

Since the pressure distribution is now known, the face velocities for each boundary can be calculated. Figure 34 illustrates the nomenclature for the face velocities.

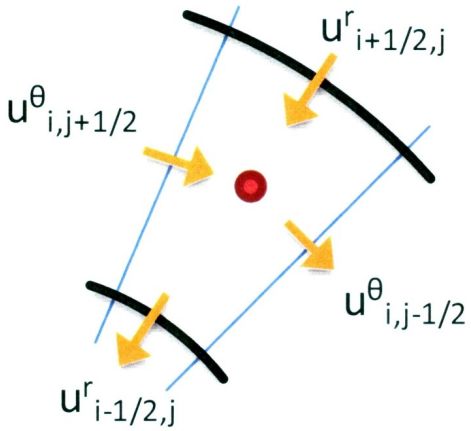


Figure 34 - Polar Face Velocity Notation

Note that the velocities mentioned throughout this section are Darcy velocities or volumetric fluxes. Real velocity (V) is related to Darcy velocity by:

$$V = \frac{u}{\emptyset}$$

where u is Darcy velocity and $\emptyset$ is porosity.

Time of flight, the main variable of interest in the streamline modeling procedure, is dependent upon the real velocity and not the Darcy velocity. Therefore, two approaches are available: one could convert at the initial stages by altering the face velocities; or once the calculations are complete the time of flight itself can be corrected. The following procedure uses the Darcy velocities, and converts to the proper time of flight once the calculation procedure is complete.

As with the Pollock Cartesian case, the face velocities for each boundary are known, but the velocity field inside is not. The next step is to determine the velocity field at all points throughout the reservoir medium through the use of interpolation. In the Pollock method

simple linear interpolation is used for this purpose. Here, the calculations will differ slightly. In the tangential direction simple linear interpolation is utilized. However, in the radial direction the velocity u_r is known to increase by the inverse of the radius, leading to much higher velocities in the near wellbore area. Therefore, to try and achieve additional accuracy, the velocity field inside a grid block is established as follows:

$$u_r(r) = \frac{a_r}{r} + b_r$$

$$u_\theta(\theta) = a_\theta \theta + b_\theta$$

In the θ-direction velocity is considered linear, while in the radial direction velocity is inversely proportional to radius. The known face velocities are used as boundary conditions to determine the unknown coefficients a_r, b_r, a_θ and b_θ, which will differ for each grid block.

3.6.3 Velocity Field Coefficients

As stated, in the θ-direction simple linear interpolation is utilized in the form of:

$$u_\theta(\theta) = a_\theta \theta + b_\theta$$

Implementing the boundary conditions at the faces velocities for a single grid block results in:

$$u^\theta_{i,j-1/2} = a^\theta_{i,j}\theta_{j-1/2} + b^\theta_{i,j}$$

$$u^\theta_{i,j+1/2} = a^\theta_{i,j}\theta_{j+1/2} + b^\theta_{i,j}$$

This is a system of two equations and two unknowns. Since the equations are linear they may be solved via matrix inversion, as follows:

$$\underbrace{\begin{pmatrix} \theta_{j-1/2} & 1 \\ \theta_{j+1/2} & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_{i,j}^{\theta} \\ b_{i,j}^{\theta} \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} u_{i,j-1/2}^{\theta} \\ u_{i,j+1/2}^{\theta} \end{pmatrix}}_{B}$$

$$X = A^{-1}B$$

$$\begin{pmatrix} a_{i,j}^{\theta} \\ b_{i,j}^{\theta} \end{pmatrix} = \begin{pmatrix} \theta_{j-1/2} & 1 \\ \theta_{j+1/2} & 1 \end{pmatrix}^{-1} \begin{pmatrix} u_{i,j-1/2}^{\theta} \\ u_{i,j+1/2}^{\theta} \end{pmatrix}$$

There are four coefficients, a_θ, b_θ, a_r and b_r, for each block. Therefore for programming purposes the values must be stored in their own individual matrix of the same (i,j) format.

The solution can also be achieved through a simple substitution and elimination technique. This option results in:

$$a_{i,j}^{\theta} = \frac{u_{i,j-1/2}^{\theta} - u_{i,j+1/2}^{\theta}}{\theta_{j-1/2} - \theta_{j+1/2}}$$

$$b_{i,j}^{\theta} = u_{i,j+1/2}^{\theta} - a_{i,j}^{\theta}\theta_{j+1/2}$$

Once $a_{i,j}^{\theta}$ is determined, it can be substituted into the second equation to determine $b_{i,j}^{\theta}$. Both methods should result in the same answer, so the appropriate choice is a matter of personal preference. Both are used in this research to provide a backup check for potential programming errors.

In the radial direction an inverse relationship is appropriate since the velocity increases significantly as fluids approach the wellbore. The equation is of the following form:

$$u_r(r) = \frac{a_r}{r} + b_r$$

Applying the appropriate boundary conditions at the radial faces results in:

$$u^r_{i-1/2,j} = \frac{a^r_{i,j}}{r_{i-1/2}} + b^r_{i,j}$$

$$u^r_{i+1/2,j} = \frac{a^r_{i,j}}{r_{i+1/2}} + b^r_{i,j}$$

This is a nonlinear system of two equations and two unknowns. It may be solved via a substitution and elimination approach, resulting in the following:

$$a^r_{i,j} = \frac{r_{i-1/2} r_{i+1/2}}{r_{i+1/2} - r_{i-1/2}} \left(u^r_{i-1/2,j} - u^r_{i+1/2,j}\right)$$

$$b^r_{i,j} = u^r_{i+1/2,j} - \frac{a^r_{i,j}}{r_{i+1/2}}$$

Once $a^r_{i,j}$ is determined it can be substituted into the second equation to determine $b^r_{i,j}$.

3.6.4 Particle Travel Time

Once the velocity coefficients are determined the velocity field throughout a grid block, and the entire reservoir medium, is known. Next, travel time of particles must be determined.

In the radial direction, the velocity $u_r(r)$ is used to determine the time required for a particle to travel to the opposite radial face. To achieve this, the integral of the inverse of velocity with respect to distance is required, as follows:

$$dr = u_r(r)dt$$

$$dt = \frac{1}{u_r(r)}dr$$

$$t_r = \int_{r_{ex}}^{r_{en}} \frac{dr}{\frac{a_r}{r} + b_r}$$

Note the order of the integration limits is chosen such that the expression results in a positive value. The alternation is due to the exit face having a smaller value than the entrance face in a polar coordinate system. Evaluating the interval $[r_{ex}, r_{en}]$ results in:

$$t_r = \frac{1}{{b_r}^2}\left(b_r(r_{en} - r_{ex}) - a_r ln\left(\frac{a_r + b_r r_{en}}{a_r + b_r r_{ex}}\right)\right)$$

The special case of a homogenous reservoir must also be considered. For the homogenous case flow is radial, with straight streamlines flowing in the direction of the wellbore. As a result, velocity will increase radially proportional to the inverse of the radius with no leakage to either side. Therefore the initial radial velocity equation:

$$u_r(r) = \frac{a_r}{r} + b_r$$

has a value of $b_r = 0$. The equation simplifies to:

$$u_r(r) = \frac{a_r}{r}$$

This is an equation with one unknown. Using the boundary condition at the outer radial face of an individual grid block, the value of the coefficient becomes:

$$a_{i,j}^r = u_{i+1/2,j}^r r_{i+1/2}$$

Using the velocity $u_r(r)$ the time required for a particle to travel to the opposite radial face is now calculated as:

$$dr = u_r(r)dt$$

$$dt = \frac{1}{u_r(r)}dr$$

$$t_r = \int_{r_{ex}}^{r_{en}} \frac{dr}{\frac{a_r}{r}}$$

$$t_r = \frac{1}{a_r}\int_{r_{ex}}^{r_{en}} r dr$$

Evaluating the interval $[r_{en}, r_{ex}]$ results in:

$$t_r = \frac{1}{2a_r}(r_{en}^2 - r_{ex}^2)$$

Once the radial time is determined, the tangential time must also be established. The time needed for a particle to travel from the entrance point to a tangential face is:

$$t_\theta = \int_{\theta_{en}}^{\theta_{ex}} \frac{d\theta}{a_\theta \theta + b_\theta}$$

Evaluating the interval $[\theta_{en}, \theta_{ex}]$ results in:

$$t_\theta = \frac{1}{a_\theta}\left(ln(b_\theta + a_\theta \theta_{ex}) - ln(b_\theta + a_\theta \theta_{en})\right)$$

The actual time for the particle to travel from the entrance face to the exit face, known as the time of flight, is the least of t_r or t_θ. This is represented by:

$$t = \min(t_r, t_\theta)$$

For the radial travel time t_r, the exit face will always be the internal radial face since the nature of the boundary conditions impose the differential pressure to be positive in this direction. For the tangential travel time t_θ, the choice of the exit boundary location is not as simple. Several cases of the exit boundary location will exist depending upon the directions of the two tangential face velocities for a block. Figure 35 provides a graphical display of the tangential boundary exit options.

Four cases are highlighted in the figure. In the first case both tangential face velocities are in the positive direction, resulting in an exit face at $\theta_{j-1/2}$. In the second case both face velocities are facing outwards, away from the block. Here, the exit boundary will depend upon the entrance location of the particle in the block. Through linear interpolation an angle must be calculated (θ^*) at which the velocity is zero. If the entrance angle is greater than θ^*, the exit boundary is at $\theta_{j+1/2}$. Likewise, if the entrance angle is less

than θ^*, the exit boundary is at $\theta_{j-1/2}$. In the third case both tangential face velocities are in the negative direction, resulting in an exit face at $\theta_{j+1/2}$.

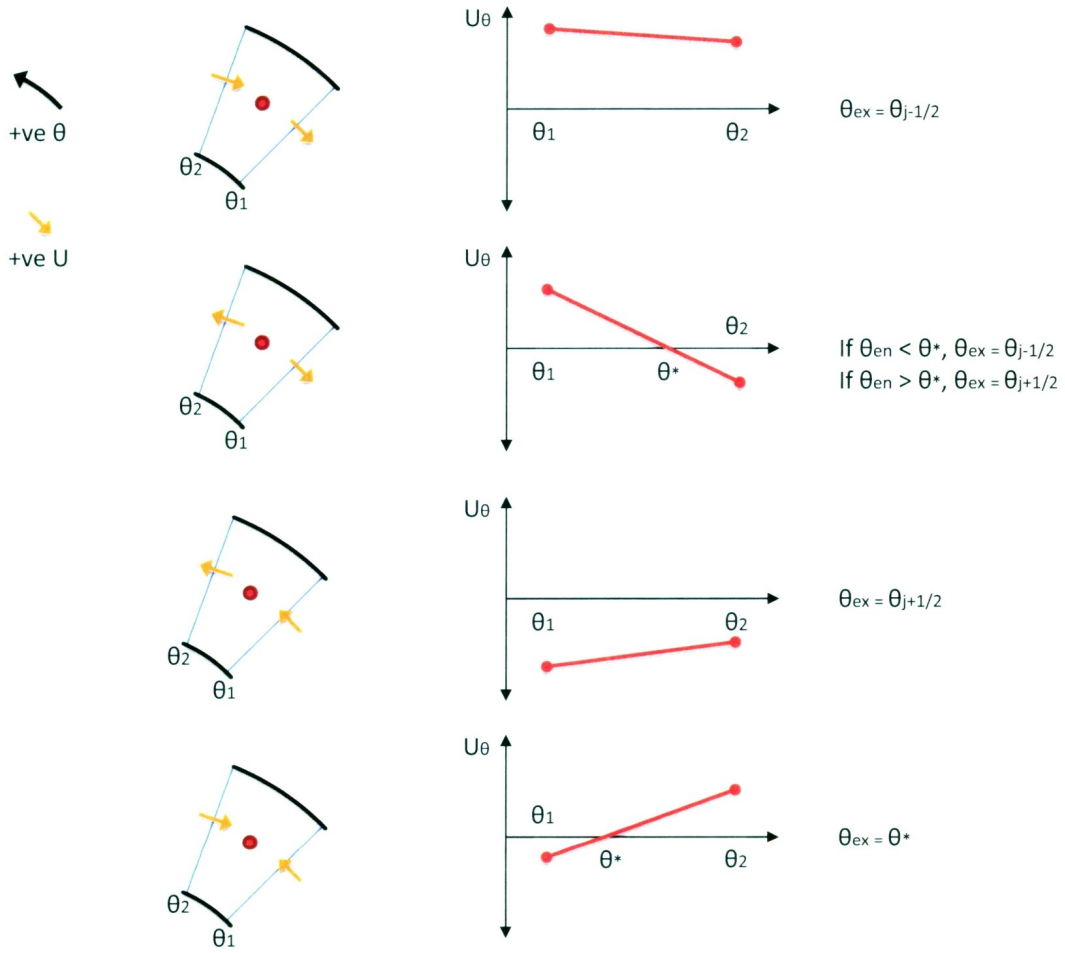


Figure 35 - Tangential Boundary Exit Options

Finally, in the last case both tangential velocities are facing inwards. As with case two, linear interpolation must be performed to determine the angle θ^* at which the velocity is equal to zero. This angle is the location to calculate the tangential travel time, since a particle will move inwards toward this point and upon reaching it will stay at this location (moving radially inwards). Several more cases also exist in the event of a face velocity being equal to zero. In these cases the tangential movement of the particles will depend

solely upon the direction of the other face velocity, moving in the direction indicated by the sign (positive or negative).

3.6.5 Particle Exit Location

In the previous section the time of flight calculation was discussed. Times were determined for the radial and tangential directions. In calculating these times the potential exit face was first determined, followed by the time it would take to reach the face. The actual boundary hit is the one which had the smallest time of flight.

Now that the exit face is known, the actual exit location along the face must be determined. This is performed by inserting the minimum time into the opposite direction time equation and solving for the distance travelled in the given time frame. The calculation is different depending upon which direction, radial or tangential, has the minimum time of flight.

If the radial time (t_r) for block (i,j) is the minimum time (t_{min}), the exit radius (r_{ex}) must equal to $r_{i-1/2}$, thereby entering grid block $(i-1,j)$. To determine the exact exit location t_{min} must be entered into the tangential time equation to calculate the resultant exit location. This is performed as follows:

$$t_{min} = t_\theta = \frac{1}{a_\theta}\left(ln(b_\theta + a_\theta\theta_{ex}) - ln(b_\theta + a_\theta\theta_{en})\right)$$

$$t_{min}a_\theta = ln\left(\frac{b_\theta + a_\theta\theta_{ex}}{b_\theta + a_\theta\theta_{en}}\right)$$

$$\frac{b_\theta + a_\theta\theta_{ex}}{b_\theta + a_\theta\theta_{en}} = e^{t_{min}a_\theta}$$

$$\theta_{ex} = \frac{e^{t_{min} a_\theta}(b_\theta + a_\theta \theta_{en}) - b_\theta}{a_\theta}$$

If the tangential time (t_θ) for block (i,j) is the minimum time (t_{min}), the exit angle (θ_{ex}) must equal to $\theta_{j\pm1/2}$, thereby entering grid block $(i, j \pm 1)$ depending upon which direction the velocity field is sending the particle. To determine the exact exit location t_{min} must be entered into the radial time equation to calculate the resultant exit radius. For a non-radial case with $b_r \neq 0$, the equation is as follows:

$$t_{min} = t_r = \frac{1}{{b_r}^2}\Big(b_r(r_{en} - r_{ex}) - a_r\big(ln(a_r + b_r r_{en}) - ln(a_r + b_r r_{ex})\big)\Big)$$

$$t_{min} = \frac{1}{b_r}(r_{en} - r_{ex}) - \frac{a_r}{{b_r}^2}\big(ln(a_r + b_r r_{en}) - ln(a_r + b_r r_{ex})\big)$$

This must be solved via a numerical method, such as Newton-Raphson. First, rearranging the function to set it equal to zero:

$$f(r_{ex}) = t_{min} - \frac{1}{b_r}(r_{en} - r_{ex}) + \frac{a_r}{{b_r}^2}\big(ln(a_r + b_r r_{en}) - ln(a_r + b_r r_{ex})\big) = 0$$

Taking the derivative of the function, also equal to zero:

$$f'(r_{ex}) = \frac{d}{dr_{ex}} f(r_{ex}) = 0 + \frac{1}{b_r} - \frac{a_r}{{b_r}^2}\frac{b_r}{a_r + b_r r_{ex}} = 0$$

$$f'(r_{ex}) = \frac{1}{b_r} - \frac{a_r}{b_r}\frac{1}{a_r + b_r r_{ex}} = 0$$

The general Newton-Raphson equation is:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Applying this equation to the current scenario results in:

$$r_{ex:\,n+1} = r_{ex:\,n} - \frac{f(r_{ex:\,n})}{f'(r_{ex:\,n})}$$

$$r_{ex:\,n+1} = r_{ex:\,n} - \frac{t_{min} - \frac{1}{b_r}(r_{en} - r_{ex}) + \frac{a_r}{{b_r}^2}\left(ln(a_r + b_r r_{en}) - ln(a_r + b_r r_{ex})\right)}{\frac{1}{b_r} - \frac{a_r}{b_r}\frac{1}{a_r + b_r r_{ex}}}$$

This is an iterative procedure which continues until $r_{ex:\,n+1}$ approaches $r_{ex:\,n}$ to within a chosen limit.

For the special radial case of a homogenous media, b_r has dropped out of the initial expression. In this scenario t_{min} cannot equal t_θ since there is no tangential movement. Therefore, the Newton-Raphson method will not be necessary.

In most cases the Newton-Raphson method is a stable approach which quickly converges to a solution. However, in a small number of cases which experience significant tangential movement the values of the velocity coefficients may force the subject of the natural logarithm within the expression to be negative upon successive iterations. As a result, the Newton-Raphson will return an imaginary number.

To combat this error a bisection method is utilized. A bisection method is another numerical technique capable of solving this expression. The method is slower than the Newton-Raphson, but always converges to a solution.

Recall that the need for this arises when t_{min} is equal to t_θ, resulting in a need to solve for r_{ex} is the following expression:

$$t_{min} = \frac{1}{b_r}(r_{en} - r_{ex}) - \frac{a_r}{{b_r}^2}\big(ln(a_r + b_r r_{en}) - ln(a_r + b_r r_{ex})\big)$$

This may be rearranged as a function equaling zero, as follows:

$$f(r_{ex}) = t_{min} - \frac{1}{b_r}(r_{en} - r_{ex}) + \frac{a_r}{{b_r}^2}\big(ln(a_r + b_r r_{en}) - ln(a_r + b_r r_{ex})\big) = 0$$

In the bisection method two initial guesses are required; each on opposite sides of solution. To ensure both are on opposing sides they must each be evaluated in the function, with their product resulting in a negative value. This is represented as:

$$f\big(r_{ex:guess\,1}\big)f\big(r_{ex:guess\,2}\big) < 0$$

In this scenario the initial guesses are clear. The block in which the iteration is taking place is known. Therefore the initial guesses are the blocks external and internal radii.

Once the guesses are confirmed, a new guess in created halfway between the initial guesses, at the midpoint. This is represented as:

$$r_{ex:n} = \frac{r_{ex:guess\,1} + r_{ex:guess\,2}}{2}$$

The value of the expression is then calculated at the midpoint. The product of the function evaluated at the midpoint with the value of the function evaluated at the upper or lower bounds that results in a negative indicates the region in which the solution must lie. Focusing solely on the lower bound, if $f(r_{ex:n})f(r_{ex:lower}) < 0$, the solution must lie

within the lower bound. If $f(r_{ex:n})f(r_{ex:lower}) > 0$, the solution must lie within the upper bound.

Knowing the new region in which the solution must lie, a new midpoint is calculated and the process is continued until the relative error between successive iterations is less than a chosen limit. Note that this technique, while accurate, is more numerically intensive than the Newton-Raphson, therefore, it is only used in circumstances in which Newton-Raphson does not converge.

As discusses in Section 3.6.2, the time of flight calculated throughout this procedure is not the actual time of flight since it has been calculated using Darcy velocity as oppose to real velocity. Since the proper conversion involves a multiplication by a constant for both directions (radial and tangential) it will not affect the calculation of exit points explained throughout this section. But to ensure the proper flow rate can be determined, to be discussed in Section 3.8, the time of flight must be converted to the actual travel time a particle will experience. This is performed as follows:

$$t_{actual} = t_{calculated}\emptyset$$

where $\emptyset$ is the porosity of the grid block.

3.7 Streamline Modeling: General Case Study Results

The following sections show the results of the streamline models for the same three generalized case studies presented in Section 3.5. These are a homogenous reservoir (case A), a reservoir with a large heterogeneity of low permeability (case B) and a reservoir with a large sector of high permeability (case C). Again, these cases are not meant to be realistic scenarios. They are provided to portray the success of the modeling technique as well as enhance the knowledge gained from the streamline modeling procedure.

3.7.1 Case A: Homogenous Reservoir

The details of the specific modeling parameters are provided in Table 1 in Section 3.5.1. Recall the case is that of a homogenous reservoir. Figure 36 provides a polar plot of the streamlines for this case.

The figure represents the results expected from theory. Due to the constant permeability throughout the reservoir there are no heterogeneities to cause disruptions in the pressure distribution. As a result, all streamlines move radially inward from the external boundary to the well. Figure 37 provides a close-up of the near well region. Note that very slight waviness can be seen. However, this is a function of the plotting within MATLAB and not an error within the code. No tangential movement is observed for the streamlines in this case.

As can be seen in Figure 37, all streamlines remain radial to the final point of destination, the wellbore. At the wellbore, the streamlines are terminated since new flow regimes will take place within the wellbore with the fluid progressing up the well.

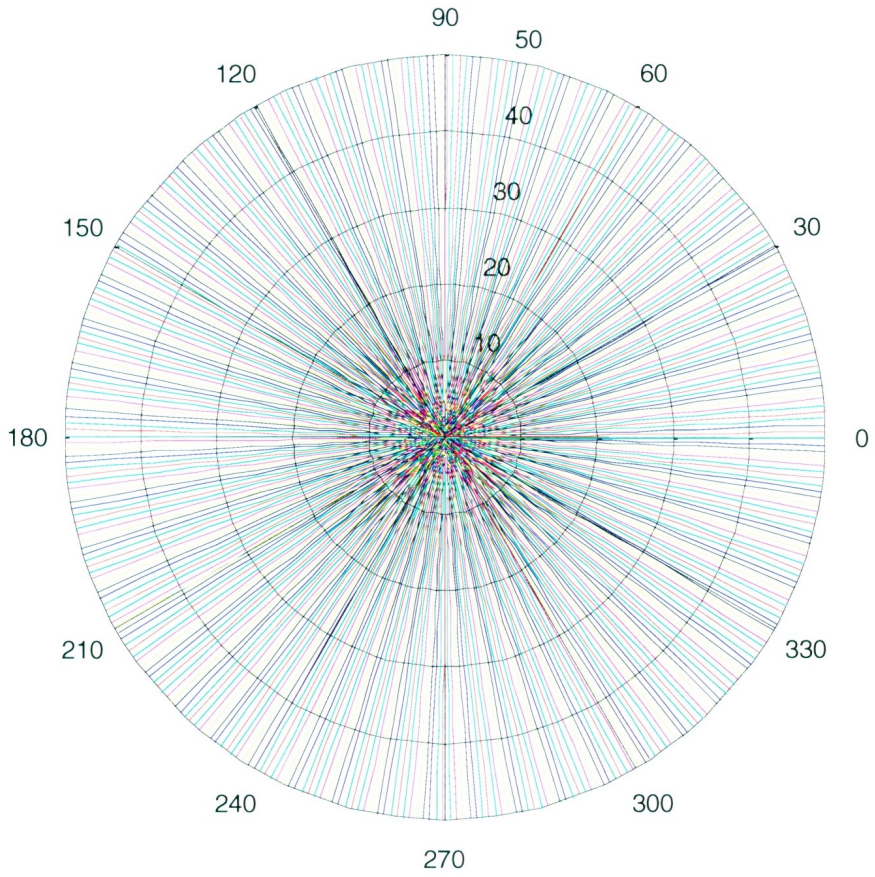


Figure 36 - Case A: Streamline Distribution

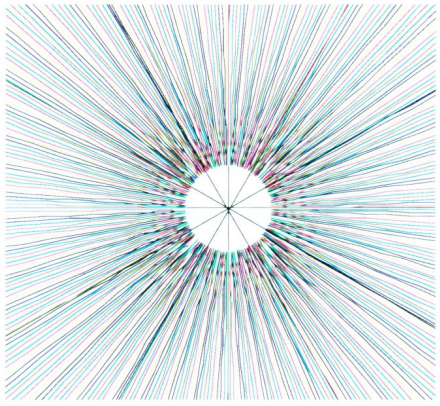


Figure 37 - Case A: Close-up (8x) of Radial Streamlines

3.7.2 Case B: Heterogeneity with Low Permeability

The details of the specific modeling parameters are provided in Table 2 in Section 3.5.2. Recall the case is that of a heterogeneous reservoir with a block of low permeability in the upper left quadrant. Figure 38 provides a polar plot of the streamlines for this case.

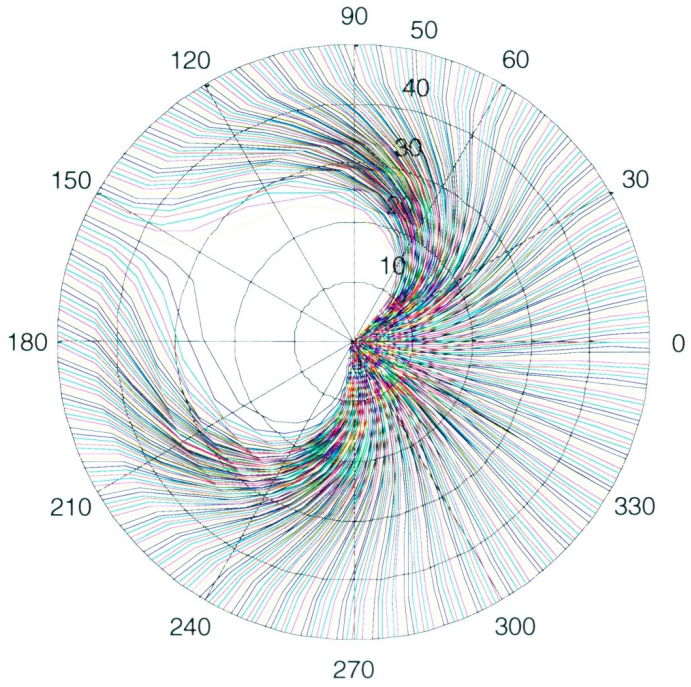
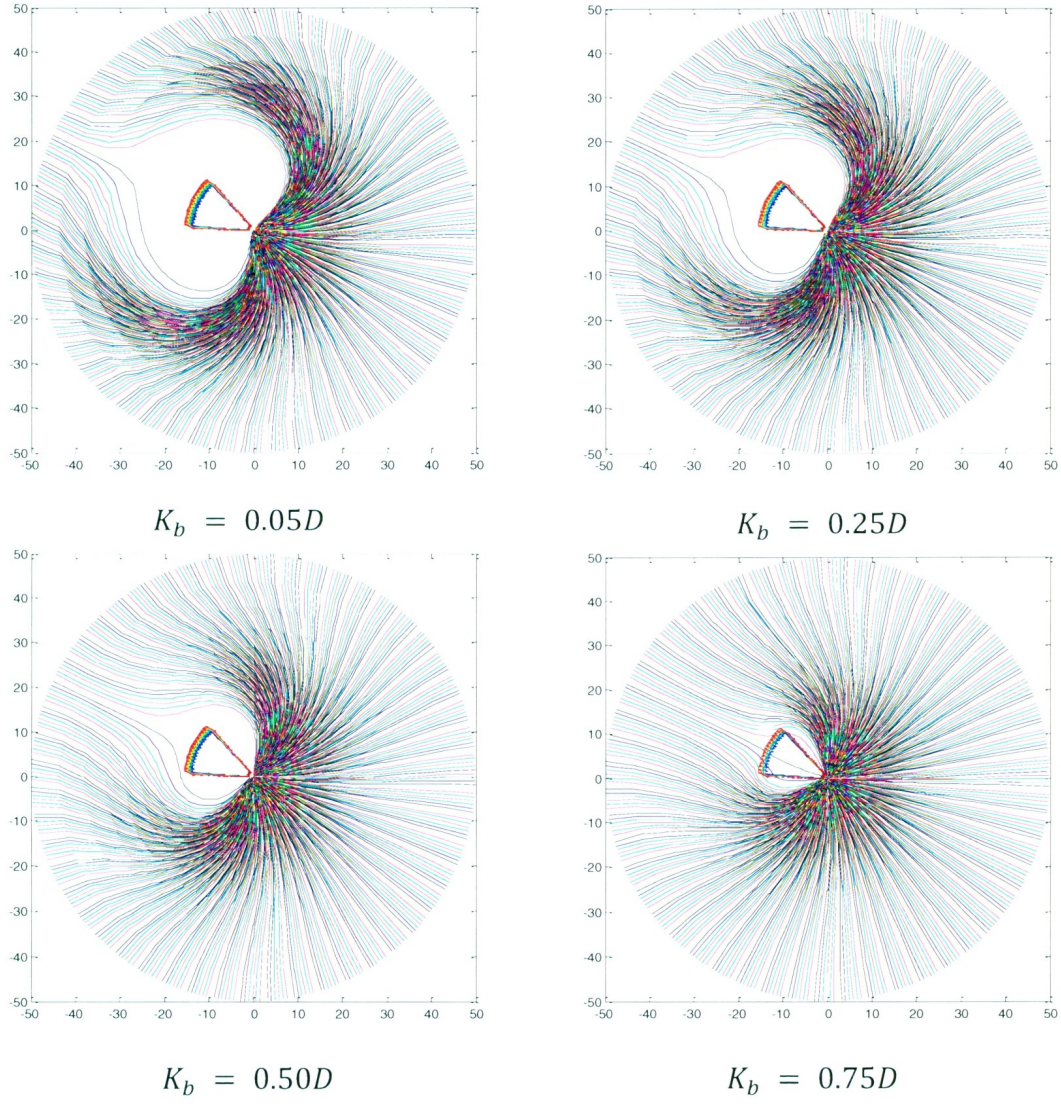


Figure 38 - Case B: Streamline Distribution

As can be seen in the figure, significant curvature exists for the majority of the streamlines. The streamlines tend to deflect away from the area of low permeability in the exterior portions of the reservoir. Once past the area of low permeability, the streamlines deflect back towards the wellbore. Note that the streamlines with the most curvature are those closest to the heterogeneity, while the curvature diminishes with distance away from the heterogeneity. Also note that 180 degrees opposite to the center of the heterogeneity (at approximately 330 degrees in the figure), the influence of the low

permeability area is minimal, allowing the streamlines to flow radially from the external boundary to the wellbore.

The degree to which the streamlines deflect away from the heterogeneity is a strong function of the permeability of the heterogeneity. Figure 39 demonstrates the effect of permeability on the streamline distribution. Four figures are provided altering the heterogeneity permeability from 0.05 Darcy to 0.75 Darcy.



$K_b = 0.05D$ $K_b = 0.25D$

$K_b = 0.50D$ $K_b = 0.75D$

Figure 39 - Case B: Effect of Block Permeability

The figure shows the permeability of the heterogeneity has a strong effect on the nature of the streamlines. When the permeability is very low the streamlines deflect far away from the heterogeneity. As the permeability increases the deflection and curvature reduce, causing the streamlines to move closer to the heterogeneity. When the block permeability is relativity close to the bulk permeability the streamlines experience only slight curvature, allowing some of the streamlines to pass through the heterogeneity.

The location and size of the heterogeneity will also have an effect on the nature of the streamlines. Figure 40 shows how changes in both block size and location affect the streamline distribution.

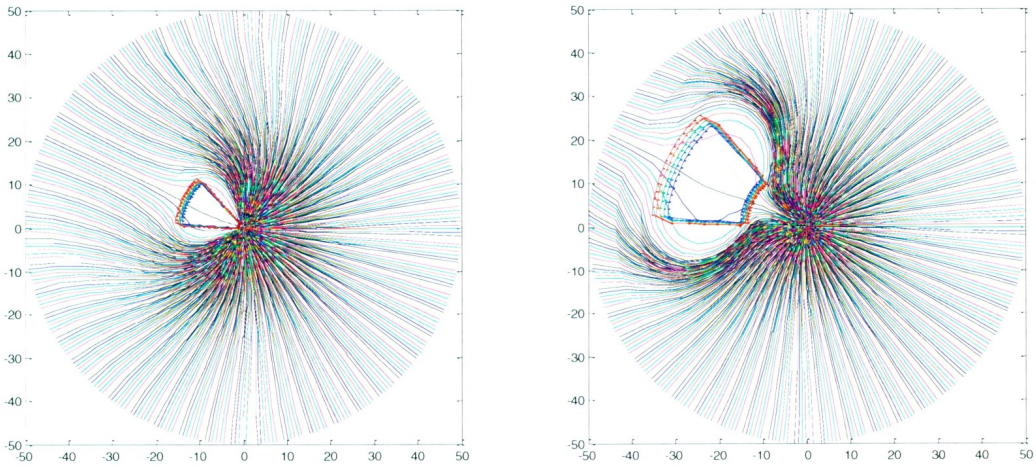


Figure 40 - Case B: Effect of Block Size and Location

In the left subfigure a heterogeneity near the wellbore with a permeability of 0.75 Darcy is shown. In the subfigure to the right the heterogeneity is moved further from the wellbore. The block still covers the same number of tangential sectors, but since arc length increase with distance from the wellbore the overall size of the block is increased.

As can be seen in both figures, the streamlines tend to deflect around the block of low permeability with very few traveling through the block. The figure to the right shows the convergence of the streamlines once the heterogeneity is passed. The streamlines begin to curve back inward, sweeping the fluid in the entire region around the wellbore.

3.7.3 Case C: Fractured Sector with High Permeability

The details of the specific modeling parameters are provided in Table 3 in Section 3.5.3. Recall the case is that of a heterogeneous reservoir with a large sector of high permeability in the upper left quadrant. Figure 41 provides a plot of the streamlines for this case. Note that it is now plotted in Cartesian coordinates to enhance the clarity of the figure.

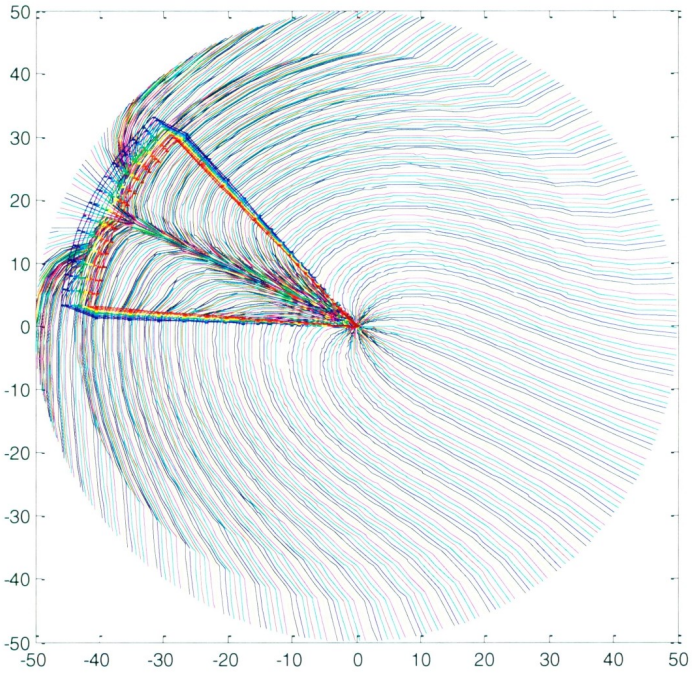


Figure 41 - Case C: Streamline Distribution

As can be seen in the figure, significant curvature exists throughout the majority of the streamlines. The very high permeability of the sector (10 Darcy) generates low differential pressure within the sector. This then causes a large differential pressure around the sector, attracting particles to the vicinity. Once inside the sector the streamlines deflect towards the wellbore.

As with the previous case, the permeability of the sector also has a large influence on the nature of the streamlines. Figure 42 provides the same case as discussed thus far with the permeability of the sector reduced to 1.5 Darcy.

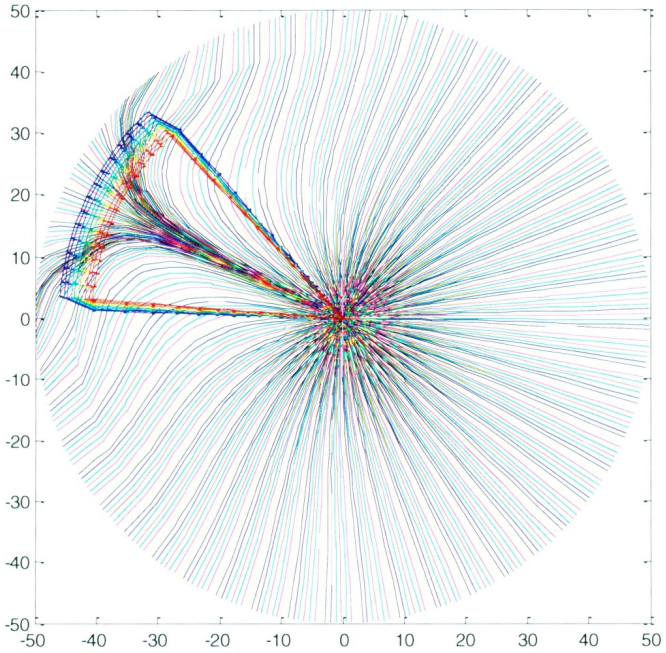


Figure 42 - Case C: Effect of Sector Permeability

The figure above shows much less curvature than the previous model. The lower differential pressure around the sector results in less of a pressure differential between the sector and the bulk reservoir. As a result, many of the streamlines flow radially into the well, while only the streamlines near the sector are significantly affected.

3.8 Determining Flow Rate from Streamline Models

3.8.1 Stream Tube Flow Rate Formulation

To determine flow rate from a streamline model the concept of stream tubes is invoked. Stream tubes involve the bundling of streamlines together such that an area is created between them. Using this area and properties of the representative streamlines, flow rate calculations are performed.

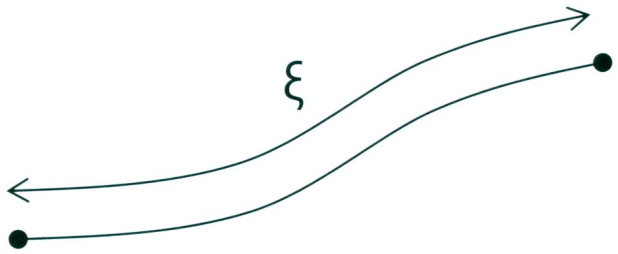


Figure 43 - Streamline Length Notation

Figure 43 provides a schematic of a single streamline with its corresponding length notation. Let ξ be the length of a streamline from an entrance point to an exit point. The velocity along the streamline is:

$$V = \frac{d\xi}{dt}$$

Rearranging to isolate the time differential:

$$dt = \frac{1}{V} d\xi$$

Integrating this results in:

$$t = \int_0^{\xi} \frac{1}{V} d\xi$$

Note that face velocities (u) referred to earlier were volumetric fluxes. Recall that volumetric flux is related to real velocity by:

$$V = \frac{u}{\phi}$$

Substituting this into the formulation:

$$t = \int_0^{\xi} \frac{\phi}{u} d\xi$$

The general relation for volumetric flux is:

$$u = \frac{q}{A}$$

where q is flow rate and A is area. Again substituting:

$$t = \int_0^{\xi} \frac{\phi A}{q} d\xi$$

In steady state conditions q is a constant, allowing it to be brought outside the integral:

$$qt = \int_0^{\xi} \phi A d\xi$$

In discretized form, summing along the segments per grid block:

$$q = \frac{1}{T}\sum_{i,j} \phi_{i,j} A_{i,j} \Delta\xi_{i,j}$$

This is the general formulation to determine the flow rate from a streamline model. The majority of these variables are known, but their application may be tricky. The following two sections provide two alternative approaches for determining the area between streamlines, and thereby provide two methods to generate flow rate. The first is a simple approximation, while the second is a rigorous and mathematically correct procedure.

3.8.2 Formulation One: Approximate Area

Determining the distance between two polar coordinate points $\left((r_1, \theta_1) \text{ and } (r_2, \theta_2)\right)$ can be determined by modifying the law of cosines, since segments are straight lines between points. The formulation results in the following:

$$\Delta\xi = \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\theta_1 - \theta_2)}$$

Figure 44 provides a schematic of the notations used in the stream tube flow rate calculation procedure.

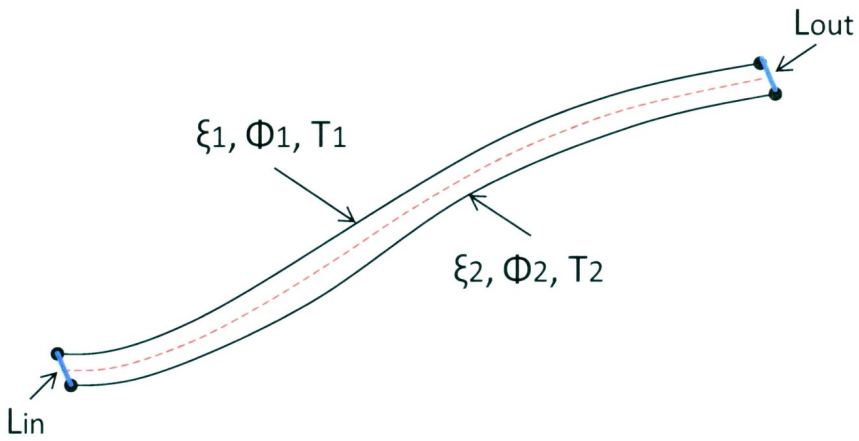


Figure 44 - Stream Tube Notation

To determine the area between two streamlines, or a stream tube, average values of the remaining parameters must be determined. Simple arithmetic averages are used, as follows:

$$T = \frac{T_1 + T_2}{2}$$

$$\xi = \frac{\xi_1 + \xi_2}{2}$$

$$\emptyset = \frac{\emptyset_1 + \emptyset_2}{2}$$

where T is the average time of flight along a stream tube segment, ξ is the average length of the stream tube segment, and $\emptyset$ is the average porosity along a stream tube segment.

The calculation of average porosity may be slightly more involved, since calculations are being done per successive radial face intersection, allowing for the possibility of a single streamline to travel through more than one tangential block with different porosities per

individual streamline. In these instances the average porosity along a single streamline must first be calculated, followed by the average for the stream tube porosity.

Following this procedure, the area calculation may be approximated as follows:

$$A = h\left(\frac{L_{in} + L_{out}}{2}\right)$$

Where h is the reservoir sector depth and L is the distance between streamlines, calculated at the entrance points and exit points on each radial intersection, using the previous law of cosines reduced since the radii are the same. This reduction leads to:

$$L = r\sqrt{2(1 - \cos(\theta_1 - \theta_2))}$$

Since the stream tube length $\Delta\xi$ is already included in the flow rate calculation procedure, this area refers to the average cross-sectional area perpendicular to the direction of flow for a given stream tube.

This procedure is only an approximation due to the nature of its formulation. In determining the volume as area multiplied by the average streamline length it assumes that all tubes are rectangles. This is a good approximation for radial flow, but it does not accurately represent curved streamlines in which heterogeneities may create angles of gridline intersections significantly astray from 90 degrees. The next section deals with correcting this issue.

3.8.3 Formulation Two: Cross Product Area

As an alternative to the prior methodology, a more rigorous approach may be used to determine the area between streamlines. It is a mathematical fact that the absolute value

or magnitude of the cross product between two Euclidean vectors is equal to the area formed by a parallelogram between them (O'Neil, 2003). This is depicted in Figure 45, with the equation following the schematic.

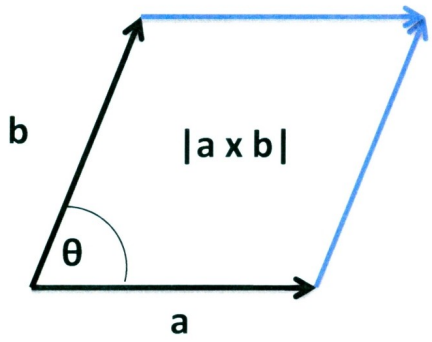


Figure 45 - Cross Product Schematic

$$A = |a \times b| = |a||b| \sin\theta$$

Using this principle, the area between streamlines can be determined. This is a more accurate approach since parallelograms are not represented as rectangles. This is especially evident when significant movement occurs in the tangential direction, forming large parallelograms as oppose to rectangles. Figure 46 provides a schematic showing the notation necessary for this procedure.

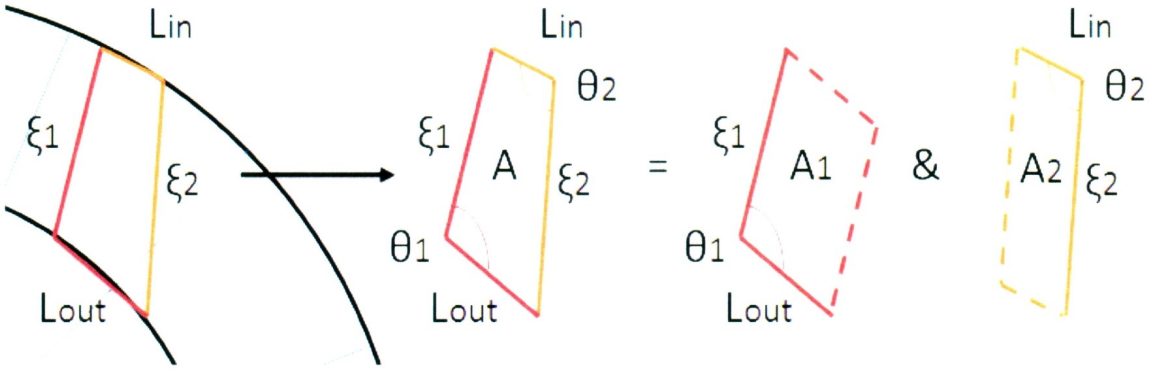


Figure 46 - Notation for Accurate Flow Rate Calculations from Streamlines

In the figure above two streamline segments on a radial grid are depicted. The lengths of each streamline segment are ξ_1 and ξ_2, respectively. Connections between the two streamlines are drawn at the radial faces, represented by L_{in} for the entrance face and L_{out} for the exit face. Note that the angles between the streamlines and the face connections are not equal to 90 degrees, thereby making the previous methodology inaccurate. Also note that opposite angles within the stream tube are not equal. Therefore two separate area calculations (A_1 and A_2) are required, followed by an averaging procedure. Finally, note that in this procedure area is the plane of the grid, not perpendicular as in the previous methodology.

To proceed with this calculation the values of the respective angles between lines are established. To do this the length of the splitting line is required. The line in question is shown in Figure 47.

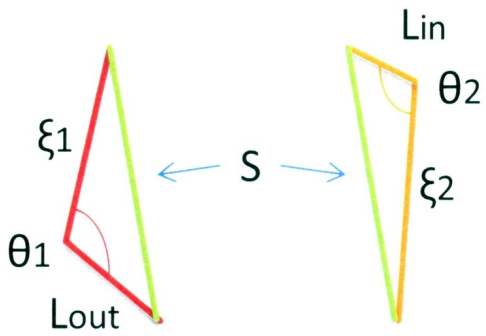


Figure 47 - Splitting Line in Stream Tube Procedure

The splitting line (S_l) is necessary to determine the respective angles. The length can be determined using the same modified cosine equation as the streamline length calculation:

$$S_l = \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\theta_1 - \theta_2)}$$

The only difference lies in the points input (along the streamline versus across the streamline). Knowing all three lengths for both triangles it is necessary to determine the desired internal angles. This is achieved using the unmodified law of cosines:

$$S_l{}^2 = \xi_1^2 + L_{out}^2 - 2\xi_1 L_{out} \cos\theta_1$$

Rearranging results in:

$$\theta_1 = \cos^{-1}\left(\frac{\xi_1^2 + L_{out}^2 - S_l{}^2}{2\xi_1 L_{out}}\right)$$

Likewise, for the second angle:

$$\theta_2 = \cos^{-1}\left(\frac{\xi_2^2 + L_{in}^2 - S_l{}^2}{2\xi_2 L_{in}}\right)$$

Knowing the angles the area can then be calculated as follows:

$$A_1 = |\xi_1||L_{out}| \sin\theta_1$$

$$A_2 = |\xi_2||L_{in}| \sin\theta_2$$

$$A = \frac{A_1 + A_2}{2}$$

The flow rate equation then becomes:

$$q = \frac{h}{T}\sum_{i,j} \phi_{i,j} A_{i,j}$$

This calculation procedure should determine an accurate flow rate within the limiting boundary conditions of the streamline models.

3.8.4 Flow Rate Verification

To verify the streamline flow rate calculation the general formulation of Darcy's law in radial geometry is utilized:

$$Q = \frac{kA}{\mu}\frac{dp}{dx}$$

Applying this to radial geometry with a thickness h and area $A = 2\pi rh$ results in:

$$Q = \frac{k2\pi rh}{\mu}\frac{dp}{dr}$$

Integrating from the external radius to the wellbore radius lead to:

$$\int_{r_w}^{r_e} \frac{dr}{r} = \frac{2\pi kh}{Q\mu}\int_{p_w}^{p_e} dp$$

$$ln\left(\frac{r_e}{r_w}\right) = \frac{2\pi kh}{Q\mu}\,(p_e - p_w)$$

Rearranging to isolate flow rate results in:

$$Q = \frac{2\pi kh(p_e - p_w)}{\mu\, ln\left(\frac{r_e}{r_w}\right)}$$

This is the general formulation of Darcy's law for a steady state homogenous medium in radial geometry. Using this formulation a comparison is made with the two methods of determining flow rate from streamlines previously discussed. Given the same conditions, the methods should generate the same values of flow, proving the validity of the method.

A general radial case is run with the case information provided in Table 4.

Table 4 – Flow Rate Verification Parameters

Case D	Radial flow in homogenous medium for verification of flow rate calculation procedure			
Factor	**Symbol**	**Name**	**Actual Units**	**Value**
H	r_w	Wellbore Radius	m	0.05
I	r_e	External Radius	m	50
J	N	Radial Blocks		50
K	M	Tangential Blocks		120
L	J	Streamlines per Block		3
M	p_{wf}	Wellbore Pressure	bar	280
N	p_e	External Pressure	bar	300
O	k	Bulk Permeability	Darcy	1
P	h	Reservoir Thickness	m	5
Q	μ	Oil Viscosity	cP	1

The outputs of the methods are provided in Table 5.

Table 5 – Flow Rate Verification Results

Unit	Area Calc. 1 (General Squares)	Area Calc. 2 (Cross Product)	Radial Darcy's Law
m^3/s	9.086×10^{-3}	9.085×10^{-3}	9.095×10^{-3}
bbl/d	4938	4937	4943

The results show both flow rate calculations procedures to be very accurate, closely approaching the value expected from the general formulation of Darcy's law within a fraction of a percent. The two procedures are close in value for a purely radial case, but will begin to differ when heterogeneities in the reservoir cause significant curvature in the streamlines. In this situation, intersection angles will stray significantly from 90 degrees, appreciably affecting the square approximation procedure. Therefore, the cross product method is used throughout this research. The accurate area calculation leads to an appropriate flow rate, as verified in the homogenous radial case previously discussed.

The MATLAB code for the near wellbore streamline model is provided in Appendix I. Please note that the entire code is novel, developed from scratch by the author of this thesis.

- Chapter 4 -

4.0 Near Wellbore Completion Modeling

4.1 Introduction

In Chapter 3 the development of a near wellbore streamline model for a two dimensional heterogeneous isotropic reservoir in polar coordinates was established. General case studies were investigated to validate the pressure solutions, streamline distributions and flow rate calculations. All cases corresponded with what would be expected from theory, thereby proving the model to be valid within the accuracy of the underlying assumptions.

In this chapter the model developed is applied to case studies that are more applicable to real industrial scenarios. The goal is to use the near wellbore model as a means to model the effectiveness of various completion strategies. A completion is the connection or interface between a well and a reservoir that promotes flow in the vicinity. With the model developed, various completions may be investigated in terms of their effectiveness, including cased and perforated wells as well as open-hole completions.

To determine the influence of variables on the response in question, Design of Experiment (DOE) methodologies are invoked. An explanation of the DOE approach is provided in Section 4.2, followed by an explanation of each of the case studies along with their respective results.

4.2 Design of Experiments Methodology

4.2.1 Design of Experiments

DOE is a methodology for systematically applying statistics to experiments, and involves the planning of experiments to ensure valid conclusions can be extracted from the results. The methods were initially invented by Ronald Fisher who discovered the analysis of variance (ANOVA) technique to objectively determine which factors were important as a result of experimentation (Fisher, 1918). Since then, numerous techniques have been developed, including factorial designs, fractional factorials designs and various response surface methodologies (RSM). The techniques have the ability to simplify systems and determine the effects for various complex models with numerous factors, including linear, interaction and polynomial behavior.

One of the main benefits of DOE is the reduced number of runs required to gain a full and accurate understanding of experimental results. The reduced runs result in less time and money being spent during experimental procedures. The methods work by changing the levels of multiple factors for each run as oppose to the traditional one factor at a time technique. It then uses methods of statistical analysis to formulate a model to represent the experiment results. This may be performed for simple experiments as well as complex scenarios involving numerical simulations. DOE methods also have the ability to determine the impact of variable interactions, a significant oversight in the one factor at a time approach.

Advanced experiment design is now being used in many fields of engineering, providing accurate and improved understanding of experiments. Various uses within the area of

reservoir engineering and simulation include the work of Faidi et al. (1996), White et al. (2001), Vanegas & Cunha (2006), Carreras et al. (2006), Kalla & White (2007), Risso et al. (2007), Amudo et al. (2008) and Ahmadloo et al. (2009). In the literature reviewed, no work has been done showing the application of response surface techniques in determining the importance of near wellbore parameters on reservoir productivity. Streamline modeling in the near wellbore region is in itself a novel technique. Therefore, the results of this procedure provide new insight into this area of research.

4.2.2 Generalized DOE Process

In general, there are three main technical objectives when performing DOE techniques. These objectives are:

- Determine which factors are significant to the experimental results;
- Determine if factor interactions are significant to the experimental results; and
- Create a response surface capable of explaining and simplifying the behavior of a complex model.

Regardless of the objective of a particular experiment, the DOE process follows a systematic procedure. Five critical steps in a generalized DOE process are depicted in Figure 48. The steps are described as follows:

1) Description of the Experiment: This step requires a general explanation of the experiment to be performed. Primarily, this involves an explanation of the factors (variables) to be investigated and the desired response(s) to be measured. This is one of two key engineering steps requiring knowledge of the experiment and proper engineering judgment.

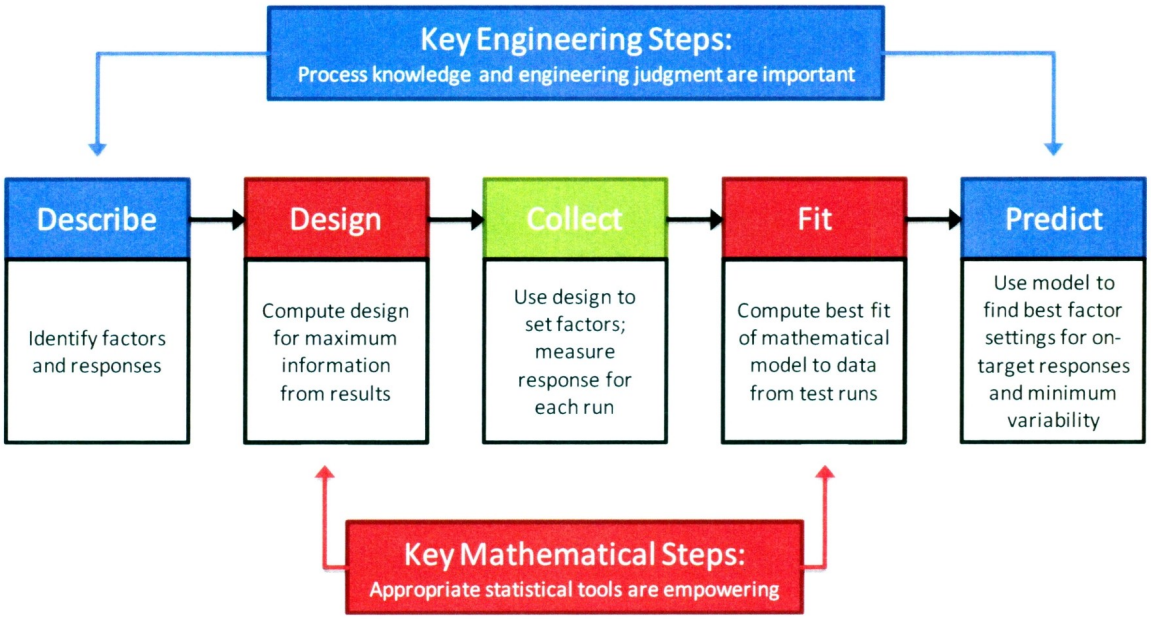


Figure 48 - Critical Steps in DOE Process

Source: Modified from (JMP, 2005)

2) Experiment Design: This step involves determining the appropriate design required for the experiment. This is the first of two key mathematical steps which involve applying the appropriate statistical tools based upon the desired objectives of the experiment. The choice of experiment design is dependent upon the desired objectives, the current understanding of the experiment and the resources required to run the experiment.

3) Data Collection: Once the appropriate design has been selected the experiment must be performed. The experiment design will generate combinations of runs based upon the chosen factors and levels (design range or design space). Each run must be performed and recorded for subsequent analysis.

4) Model Fitting: Once the experiment results are determined, the next key mathematical step involves fitting a model to the response. This is achieved

through the use of ANOVA and other statistical techniques. The process involves determining which factors are statistically significant and developing a model based upon the significant factors that accurately represent the results of the experiments. This model must not violate the underlying assumptions within the ANOVA process, including normality, constant variance and randomness of residuals.

5) Prediction: Once a valid model has been determined, it may then be used for two purposes; explaining the nature of the response and for prediction. Explaining the response involves using engineering knowledge to examining the response surface to explain the model results. Prediction involves using the mathematical model created to predict new values. This may also involve optimization of the response based upon model prediction. It is important to note that the models are only valid for local analysis. Outside of the design space the models are no longer valid.

4.2.3 Important Principles within DOE

There are various principles within DOE which contribute to its success. Some of the more prominent principles include comparison, randomization, replication and blocking.

Comparison refers to the statistical assessment of the relative importance of each factor. This is an essential component of the ANOVA process in which the mean square is utilized to estimate the variance for a particular factor. The P-value is calculated for each factor, determining the level of significance in which the null hypothesis will fail. If this value is less than the chosen significance level, the factor is considered statistically significant.

Randomization is a requirement which imposes runs to be random in any DOE analysis. Randomization ensures that systematic errors which may occur during an experiment are randomly distributed throughout the results, thereby reducing the likelihood of their incorporation in the analysis.

Replication refers to runs being repeated at the same factor levels to gain a measure of the natural variance within an experiment. While this is essential for physical experimentation, it has no value in numerical/computer experiments in which errors as a result of external sources cannot occur. Hence, in numerical simulations such as the cases studied in this research, the natural variance is zero, representing no need for replication.

Blocking is another important concept in DOE processes. It refers to a statistical technique which accounts for differences which may arise from performing experimental runs under different circumstances not under study in the experiment setup. In the event groups of runs may have to be performed at different times, by different people, under different circumstances or the like, blocking may be utilized to gain a measure of the variance which may have occurred as a result of these circumstances and remove them from the experimental results. Blocking is not required in this research.

4.2.4 DOE Techniques Utilized in this Research

4.2.4.1 Fractional Factorials

General two level factorial designs simultaneously alter the levels of factors to determine the relative importance of each factor on the response using the least number of required runs. Factors are altered over two levels, high and low, to determine their significance in

the experiment. In full factorial designs, all possible combinations of factors are run to determine their effects.

A fractional factorial design involves using a select number of runs compared to that of the full factorial. Providing a large number of factors are being investigated, a fractional factorial design can gain maximum information in a limited number of runs. A two level fractional factorial design is normally represented in the form 2^{k-n}, where '2' refers to the number of factor levels, 'k' refers to the number of factors and 'n' refers to the number of generators, or the degree of the fraction. For example, $n = 1$ represents a $1/2$ fraction design, $n = 2$ represents a $1/4$ fraction design and so on.

The fractional factorial process requires the selection of high order terms which are used as block generators to determine which runs are carried out. The effects of the generators cannot be established and therefore must be carefully selected to ensure their importance is negligible. The generators are also used to determine the alias structure of the experiment design. Alias refers to determining which factors are confounded together. In performing a fractional factorial the relative effects of factors are confounded (aliased) with other factors, meaning their individual contributions cannot be determined. Only the effect of the alias group is calculated.

Fractional factorial designs rely on the sparsity-of-effects principle. In general, this principle states that three factor interactions or higher are normally negligible, and therefore, can be assumed equal to zero. Utilizing this principle, the goal is to establish the experiment design such that individual and two factor interactions are confounded

(aliased) with negligible higher order interactions, thereby providing clear estimates of the important effects.

The resolution of a fractional factorial design is critical to its success. Resolution refers to the cumulative order of terms which are aliased together in a fractional factorial design. For example, a resolution IV design refers to an alias structure in which single factors are confounded with three factor interactions and two factor interactions are confounded with each other. According to the sparsity-of-effects principle, three factor interactions and above are normally negligible. Therefore, the most efficient fractional factorial designs are resolution V or higher. For a resolution V design, single factors are confounded with four factor interactions, while two factor interactions are confounded with three factor interactions. Therefore, providing the sparsity-of-effects principle is valid, clean estimates of all single factors and two factor interactions can be established in a resolution V design.

The main disadvantage of a two level fractional factorial design is an inability to model beyond linear and interaction behavior, though through the addition of center points the presence of curvature can be detected. If a model response shows curvature of any form a two level fractional factorial design cannot be used to create an adequate meta-model of the results. Therefore, two level fractional factorial designs are best suited for simple linear responses, or as a tool for factor screening to determine important factors. In this research, a two level fractional factorial design is used to determine the important factors for a six factor cased and perforated well completion case study, as outlined in Section 4.4.

4.2.4.2 Response Surface Methodology and IV Optimal Designs

In the event an experiment or numerical simulation has a nonlinear response, various methods of response surface methodologies (RSM) may be utilized. In general, RSM refers to "a set of techniques used in the empirical study of relationships between one or more responses and a group of variables" (Cornell, 1990 & Zangeneh et al., 2002). With respect to nonlinear responses, numerous experiment designs may be utilized to effectively match a model, or response surface, to the results.

There are numerous reasons to utilize RSM as a form of experiment design. The most common goal is to determine the factor levels required to achieve an optimization point in the response of a nonlinear model. To determine the optimal response, be it a maximum, minimum or given target, the nature of the response must be determined via some form of RSM design. RSM will create a simplified mathematical meta-model which represents the result of the experiment runs over the design space. This meta-model may then be used and analyzed to determine the factor levels required to reach the optimization goal.

Optimization and the creation of a mathematical meta-model are not always the goal of an RSM process. RSM can also be used as a tool to enhance the understanding and behaviour of complex models. In creating a response surface, the nature of how a response is affected by changes in factor levels can be determined, providing an enhanced and accurate understanding of the behaviour within the model. This is the ultimate goal of the RSM utilized in this research.

As opposed to the two level factorial designs discussed previously, RSM designs alter factors over greater than two levels. The number of levels depends upon the complexity of the nonlinear behaviour within the response. Two levels are adequate for modeling straight lines, three levels are required for quadratic equations, four levels are required for cubic equations, and so on.

The most common forms of RSM techniques are Central Composite Designs (CCD) and Box-Behnken Designs (BBD). These designs are capable of applying second order models to a response. A CCD is built-up form factorial or fractional factorial designs, while a BBD is based upon an independent quadratic design (Lye, 2009). The main differences in these models lie within the levels selected for run combinations. A CCD alters factors over five levels (only three are used for matching of a single factor), while a BBD alters factors over three levels. In addition, a BBD avoids assigning run combinations at "corner points" in which end levels of factors would be run together. This is important for experiments in which extreme combinations of factors may lead to undesirable results.

For more complicated responses, typical of those associated with complex numerical simulations, responses may be highly nonlinear. These simulations require higher order models than can be generated by a CCD or BBD. Numerous advanced designs are now available for obtaining response surfaces for complicated nonlinear behaviours. The specifics of each design will vary, but certain characteristics are required throughout all available techniques. In general, quality experiment designs must ensure factors levels and run combinations are generated throughout the design space to provide adequate information on the nature of the response. This ensures any regions with complex

changes in the response are not neglected. In addition, the fitted values generated by the meta-model must be as close as possible to that of the measured values. The best designs are sequential and require a minimum number of runs. Sequential refers to the ability to augment the experiment design to add additional complexity (more points) if higher order models are required. The design techniques must also allow for a variety of models to be developed from the data.

In this research, optimal designs are utilized to generate response surfaces for the various case studies. Optimal designs are a form of space filling designs which are utilized to capture the maximum amount of information throughout a given design space. As opposed to the set processes for level selection in BBD and CCD, optimal designs use an algorithm to select factor levels and run combinations throughout the design space. The range of the design space for each factor is chosen by the person performing the experiment and requires knowledge and proper engineering judgement. According to Stat-Ease, optimal designs are the recommended design choice when models contain both numeric and categoric factors, cubic or higher order models are required and the goal is to fit a custom model (Stat-Ease, 2011).

Several different types of optimal designs are available, including D-optimal, A-optimal and IV-optimal (also referred to as I-optimal). D-optimal designs minimize the variance associated with model coefficients; A-optimal designs minimize the average variance within the model; and IV-optimal designs minimize the prediction variance within the model (Hoos, 2003 & Stat-Ease, 2011). IV-optimal designs are utilized in this research. Note that all statistical analyses performed in this chapter are completed in Design Expert™ 8.0, created by Stat-Ease.

4.3 Model Assumptions and Representations

In the following case studies two completion types are investigated: cased and perforated completions and open-hole completions with associated filter cake buildup. A brief description of the actual completion is provided here, followed by an explanation of how the key parameters in each completion are represented within the model.

A cased and perforated well completion is an advanced completion commonly used throughout reservoirs worldwide. A schematic of a cased and perforated completion is provided in Figure 49. In these completions the casing (steel tubing encompassing the wellbore) is extended throughout the entire length of the well. The casing extends to the total depth of the well, including through the pay zone which contains the reservoir fluids. The casing acts as a barrier between the well and outside formation. The casing is cemented in-place to ensure the wellbore remains stable.

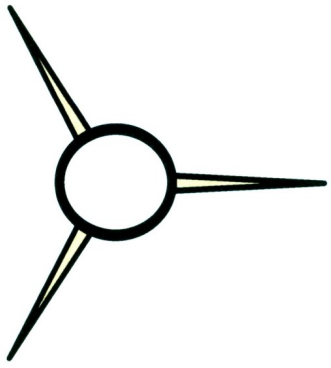


Figure 49 - Schematic of a Cased and Perforated Completion

The casing alone provides no connection between the reservoir and the well, which is essential for fluid production. To provide this connection perforations are utilized. Perforations are permeable flow channels extending from the well, through the casing and into the reservoir. They are created by running a perforating gun into the wellbore

and firing charges perpendicular to the wellbore into the pay zone. The act of creating a perforation may also cause a damage zone of reduced permeability surrounding the perforation as a result of local crushing associated with the highly charged explosion.

The purpose of the cased and perforated case studies is to investigate which variables influence the overall production of a well, which variables are most significant, and the nature of the relationship between significant parameters. This is investigated in Case Study 1 and 2 in Sections 4.4 and 4.5, respectively.

The other form of completion studied in this research is an open-hole completion. In this completion no casing or production liner is present in the pay zone connecting the reservoir and the wellbore. The zone may be left entirely bare as upon completion of drilling activities, or a means of sand control may be installed to mitigate production of fine particles. Sand control is not considered in this research.

The process of drilling an open-hole completion may lead to the buildup of a filter cake. Filter cake refers to a buildup of solid particles on the interior of the sand face of an open-hole completion. It forms as a result of over balanced drilling operations in which particles contained within the drilling mud flow into the pores of the near wellbore region and form agglomerated masses on the interior walls. Filter cake is known to have a very low permeability and as a result it can seriously affect the productivity of a well if it is not fully removed. The properties of the filter cake will vary depending upon its composition, the differential pressure exerted across the cake and the amount of shear stress experienced from the circulation of drilling mud (Lohne et al., 2009).

Once a well is completed, a cleanup process is run to remove the drilling fluid and filter cake buildup. If this process is not fully effective, filter cake may remain attached to the inner walls of the production zone.

The purpose of the open-hole completion case study is to determine the significance of the left over filter cake on the production of a well. A schematic of filter cake buildup on the wall of an open-hole completion is provided in Figure 50. Of specific interest is the impact of filter cake coverage, thickness and permeability on the productivity of a well. This is investigated in Case Study 3 in Section 4.6.

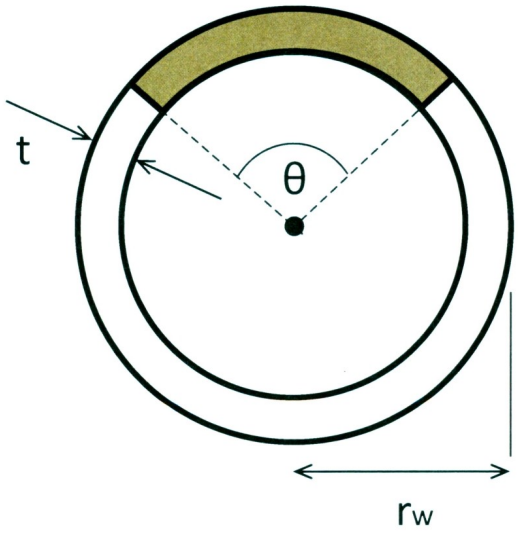


Figure 50 - Schematic of Filter Cake Buildup in an Open-hole Completion

4.3.1 Wellbore Damage Representation

To represent wellbore damage as a result of drilling activities a circular area of reduced permeability is generated around the wellbore, as depicted in Figure 51. Variables include the radius of the wellbore damage along with its corresponding reduced permeability. In Figure 51, red represents the bulk permeability of the formation, while blue represents the

region of reduced permeability. Overbalanced drilling, as required in most regulated jurisdictions, typically results in the creation of the near wellbore damage zone.

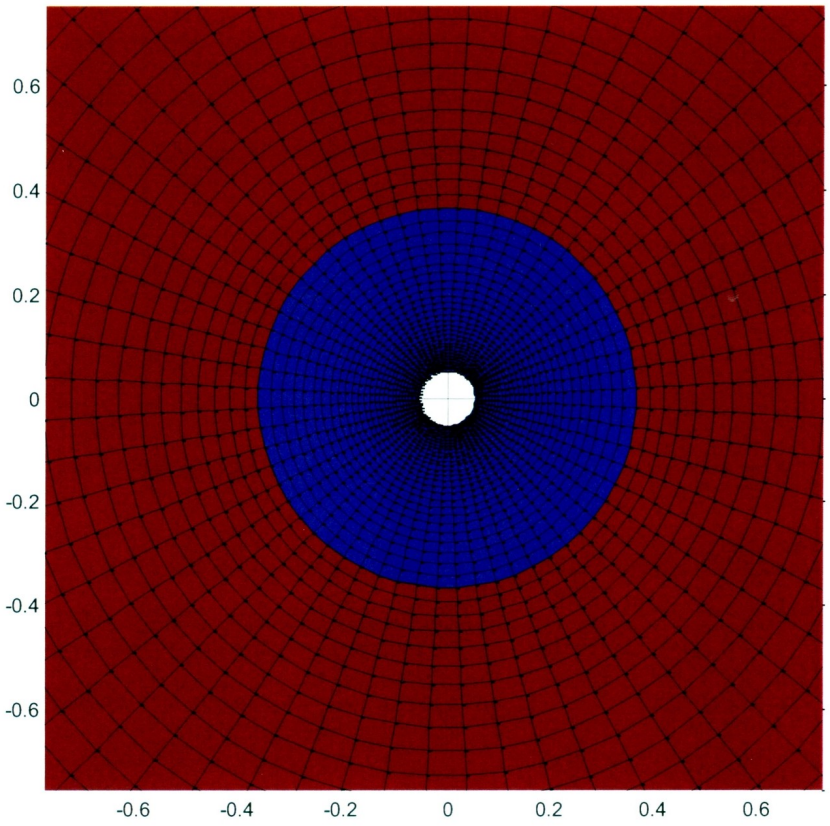


Figure 51 - Wellbore Damage Representation

4.3.2 Casing Representation

To represent the casing the internal ring around the wellbore is given an extremely low permeability, as depicted in Figure 52. This is an approximation and may not be truly representative of reality. Theoretically, however, the approach is valid. Fluid, represented by streamlines, will tend to avoid low permeability regions. The low permeability should be reflected in the pressure distribution and force streamlines to flow into the perforation. The representation of perforations within the model is explained in the next section.

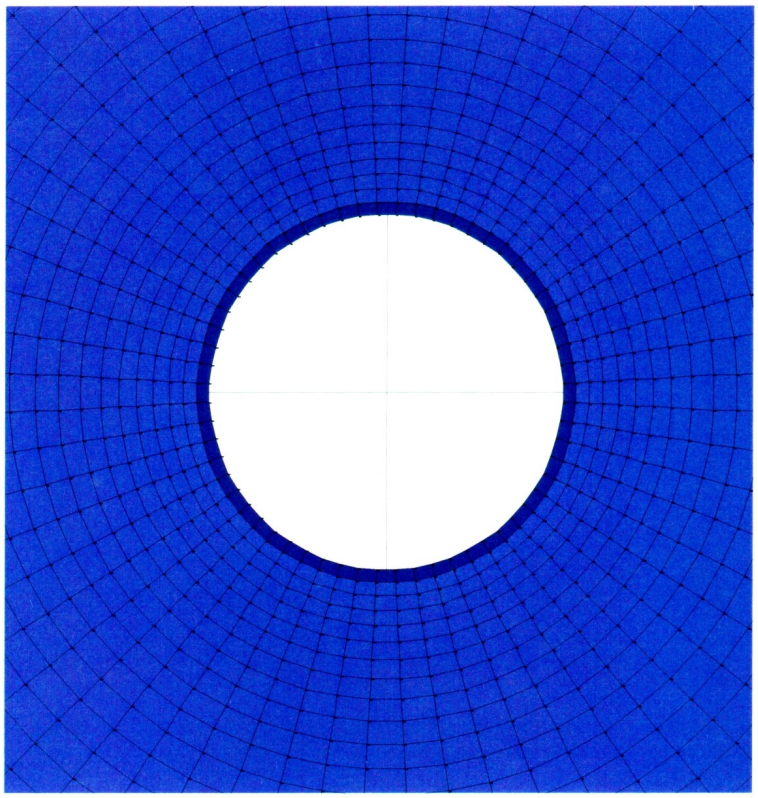


Figure 52 - Casing Representation

Successive runs testing this approximation have revealed an error in its approach. If the casing is represented as a ring of extremely low permeability around the wellbore, unavoidable numerical errors will result. The errors arise when a streamline approaches the casing. The streamlines will want to deflect from the area of low permeability, but some will wind up hitting the casing. In reality this is fine; the oil will flow around the casing and into the perforations. Unfortunately, due to the modeling procedure utilized, a streamline which comes into contact with the casing wall (external boundary of the first grid ring) must continue through the casing once hitting the boundary, representing an unrealistic scenario. The low permeability of the modeled casing causes very large travel times and effectively diminishes the flow rate of all stream tubes affected by the occurrence.

The errors occur as a result of numerical approximations. When applying the discretized solution to solve the Laplacian, pressures are calculated at each node, including the first ring nodes existing within the casing. Ideally the first node casing pressure would be equal to the pressure of the interior wellbore, representing zero differential pressure and no flow across the boundary. Unfortunately the methodology presented thus far requires a pressure to be solved at this location that is higher than the internal node pressure, resulting in flow through the casing. This error is depicted in Figure 53.

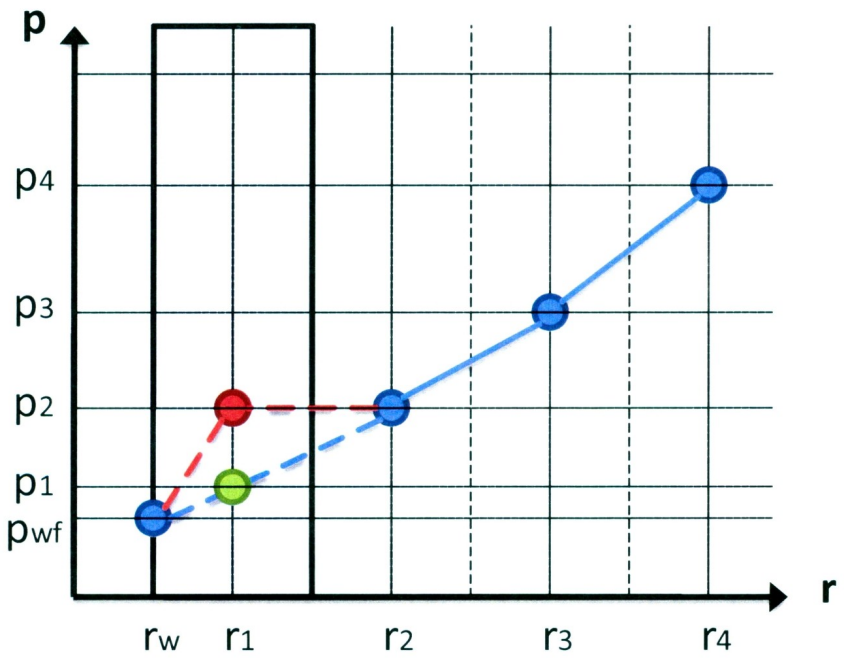


Figure 53 - Numerical Error with Respect to Casing Representation

To combat this numerical error several methods are possible. One solution is to manually set the pressure of the first node beyond the casing equal to the pressure at the casing in all areas where casing exist (no perforations in the area). This would be done after the Laplacian is solved, but before the streamlines are generated. The process would result in

zero differential pressure between radial nodes along the casing face, resulting in no driving force in the radial direction. Flow would be forced to move tangentially until reaching a perforation, at which point it would flow radially into the well.

As opposed to changing the pressure, a simpler approach with the same outcome is to set a code condition such that if a streamline hits the casing ring, the next point will be at the intersection between the casing and the nearest perforation. In this approach no alteration of the pressure or velocity distribution is required. In addition, to prevent errors in the flow rate calculations, the calculations will not include the casing ring. This is also realistic, since the casing will not contain producible hydrocarbon fluid and will not affect the results. Therefore, the streamlines are analyzed from the external boundary to the casing, as oppose to the wellbore (one additional ring).

Another solution to this problem is to go back to the original Laplacian pressure equations and set a no-flow boundary at the casing. This is a rigorous method which will provide an exact solution. The approach is discussed in Chapter 6.

4.3.3 Perforation Representation

As mentioned previously, perforations are added to allow for flow from the reservoir into the well. In addition to this, perforations should also extend beyond the formation damage zone to allow for easier flow paths into the wellbore, resulting in increased productivity.

When modeling perforations, there are several variables to consider. These include the perforation length, perforation diameter (width), number of perforations and perforation permeability. Within the model these variables can be modeled by controlling the number

of grid blocks, both radially and tangentially, that the perforations occupy, as well as the permeability of the corresponding blocks. The model representation of the perforations, along with the near wellbore damage region and casing, are shown in Figure 54. Note that the grid block color indicates different permeability in the various regions. Dark red is the highest permeability corresponding to the perforations, light red represents the permeability of the bulk formation, orange represents the permeability of the damage zone, and blue represents the permeability of the casing. A close-up highlighting the nature of the perforations connection to the well and corresponding casing is shown in Figure 55.

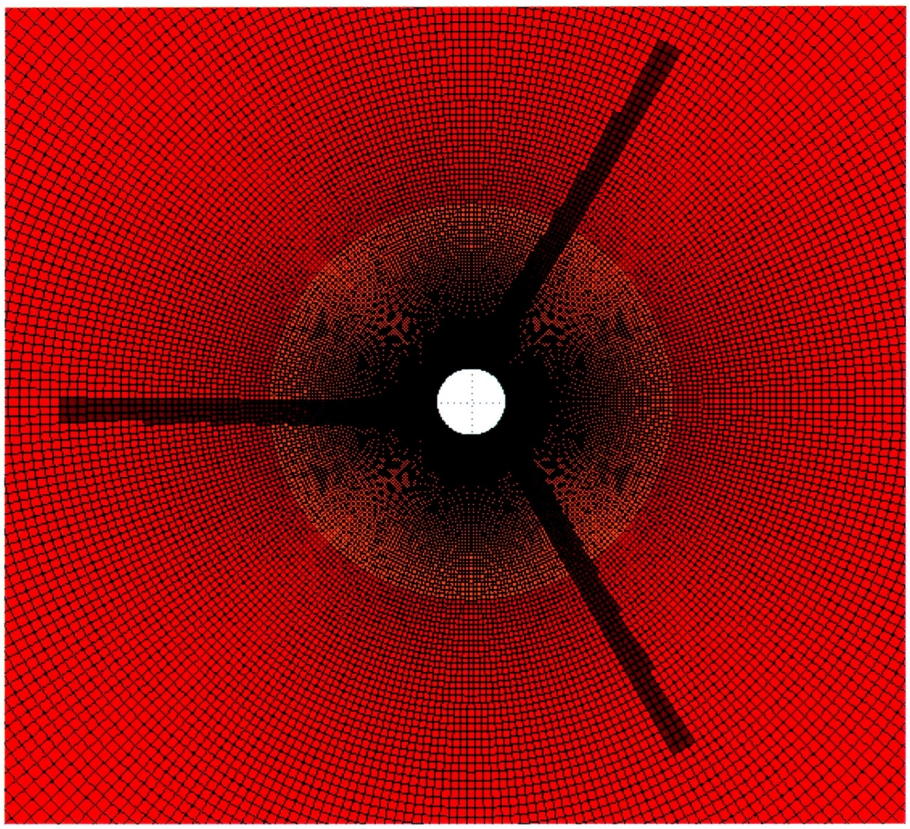


Figure 54 - Perforation Representation

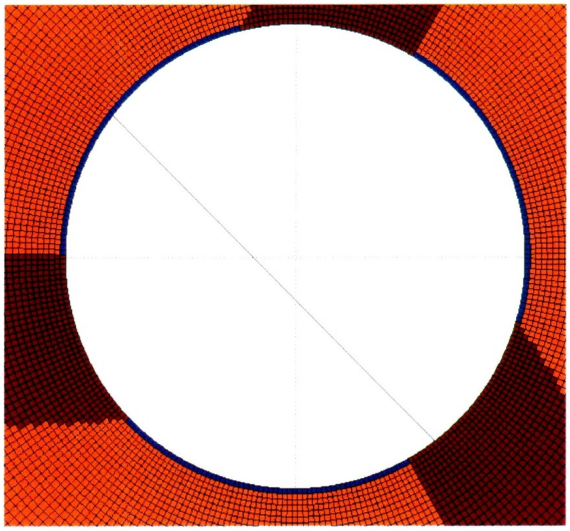


Figure 55 - Perforation Representation at Wellbore

The geometry of a perforation in a polar coordinate system is somewhat complex due to the difficulties in creating a constant width (square like) protrusion in a radial geometric system. To accommodate this issue a set perforation width is established as the input variable along with the perforation length. The perforation width is then used as a maximum perforation width. The number of tangential blocks in which a perforation occupies is required to remain less than or equal to this width for each radial step calculated. Since the width of each block grows with increasing distance from the wellbore, the number of blocks required per ring to maintain the desired perforation width will decrease. This results in the stepped look in the perforations, as observed in both previous figures. Also note that the perforation geometry has been chosen to step in one direction only.

The perforation geometry is sensitive to the grid setup. The numbers of radial and tangential blocks used in the grid are chosen in the early input stages. If an insufficient number of blocks are chosen, the perforation may exist in only one grid block. Depending

upon the chosen perforation length, this may result in significant error as a result of increasing grid size with distance. This error is depicted in Figure 56.

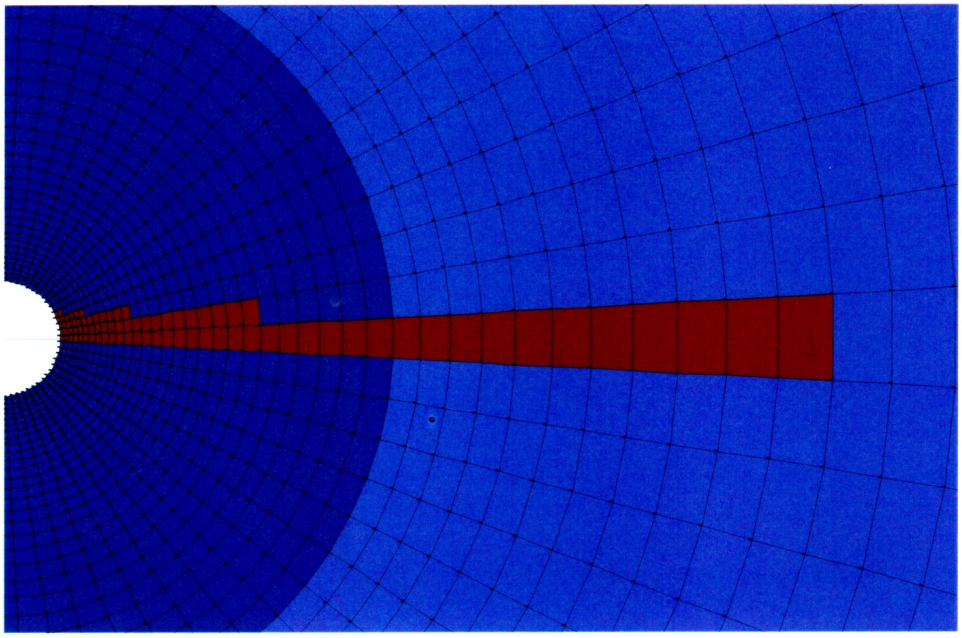


Figure 56 - Potential Perforation Error Due to Inadequate Sector Size

To prevent this error from occurring, a check must be performed to ensure the length chosen for a perforation does not have an arc length for the exterior block greater than the chosen perforation width. This check is performed following the input of variables to immediately highlight the issue. If the error occurs, an output will state its existence along with the number of tangential blocks required to prevent the error. To fix the error the number of tangential blocks must be increased to a number above the minimum level. In the case studies to follow the number of blocks chosen is well above the minimum limit in which this can occur.

4.3.4 Filter Cake and Free Space Representation

As previously discussed, filter cake is the buildup of low permeability mud on the inner walls of an open-hole completion. To represent this in the model additional rings are added inside the ring considered to be the wellbore radius. These internal grid blocks will represent both the filter cake and free space. The differentiation between the two is in their respective permeabilities. A sector of an interior ring considered to be filter cake will have a very low permeability. The remainder of the ring in which filter cake is not present is represented by a very large permeability to approximate free space. An illustration of this is provided in Figure 57. Bulk permeability is shown in green, filter cake is shown in blue, and free space is shown in red.

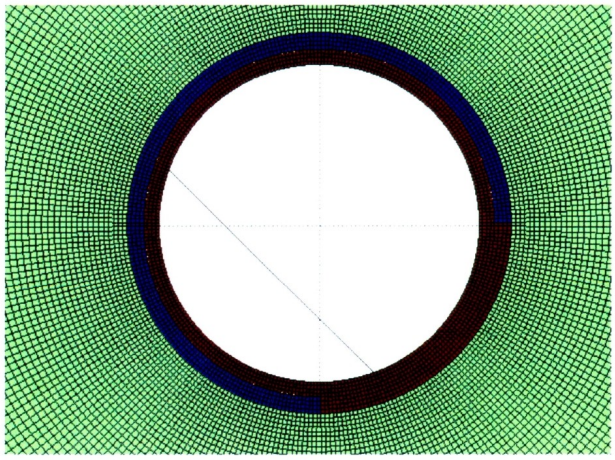


Figure 57 - Filter Cake Representation at Wellbore

Filter cake buildup is known to be thin, normally less than a centimeter thick. Therefore for adequate node spacing within this small region a large number of grid blocks is utilized. This provides for adequate grid spacing within the thin region under study.

For an enhanced understanding of the filter cake representation within the model a generalized sketch of the wellbore area is provided in Figure 58.

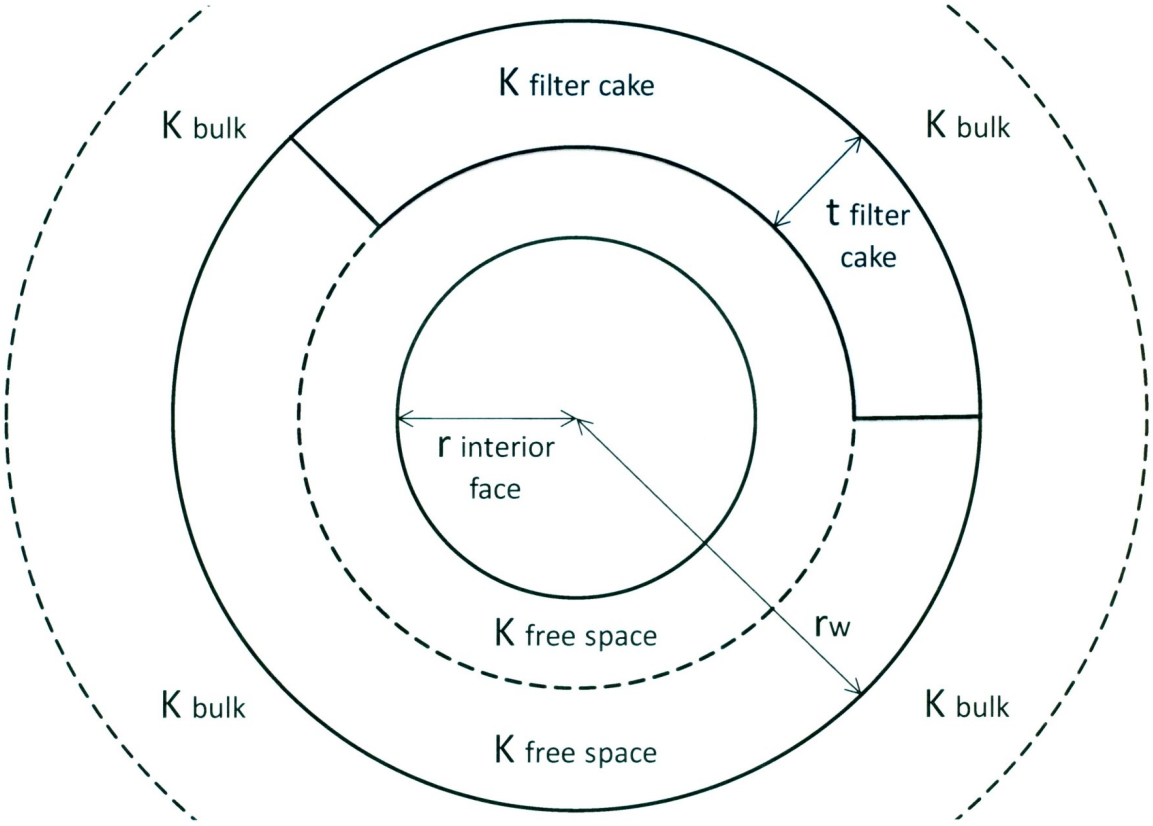


Figure 58 - Filter Cake Model Setup

In Figure 58 the wellbore radius extends beyond the first grid boundary (interior face). Within this region, filter cake is added by reducing the block permeabilities over a given thickness and angle of coverage. The remaining grid blocks (inside the wellbore radius and not associated with the filter cake) are assigned a high permeability to represent free space. In this case, the wellbore pressure does not represent the pressure at the wellbore radius. It refers to the pressure at the interior radius, with the remainder of the pressures calculated via the Laplacian process.

4.4 Case Study 1: Parametric Influences in a Cased and Perforated Well

4.4.1 Study Parameters and Assumptions

The following case study studies the main factors influencing a cased and perforated well completion. A description of the completion type is provided in the previous section along with a description of how the completion is represented in the near wellbore streamline model.

The purpose of this case study is to determine which factors influence the productivity of a well, assess the relative importance of each factor and determine whether variable interactions play a role in influencing productivity. To do this, a two level fractional factorial design is performed. Detail regarding fractional factorial designs is provided in Section 4.2.4.1.

4.4.2 Fractional Factorial Setup

In this case study, six factors (variables) are selected to investigate their influence on the productivity of a well. The factors, along with their corresponding high and low levels, are provided in Table 6.

Table 6 – Case Study 1: Factor Summary

Factor	Name	Units	Low (Actual)	High (Actual)	Low (Coded)	High (Coded)
A	Damage Zone Radius	m	0.1	0.5	-1	1
B	Damage Zone Permeability	mD	15	90	-1	1
C	Perforation Length	m	0.15	0.75	-1	1
D	Perforation Thickness	m	0.03	0.05	-1	1
E	Perforation Permeability	mD	100	1600	-1	1
F	Number of Perforations	n/a	1	3	-1	1

Note that in a two level fractional factorial design only the high and low levels of the factors are required, and runs will only be performed at these levels. The only exception is for testing curvature, in which center points are required. Also note that five of the factors are numeric, meaning they can be ranged over all values in between if required, but the number of perforations is a categoric factor. This means that only the whole values can be used. For example, 2.5 perforations is an unrealistic value and therefore assigning the factor as a categoric variable eliminates this possibility.

Numerous other factors play a role in influencing productivity. The variables not of interest in this case study are left as constants throughout each run. The values for each of the constants are provided in Table 7.

Table 7 – Case Study 1: Constants

Factor	Symbol	Name	Units	Value
G	k	Bulk Permeability	mD	100
H	r_e	External Radius	m	50
I	r_w	Wellbore Radius	m	0.05
J	p_e	External Pressure	bar	300
K	p_{wf}	Wellbore Pressure	bar	280
L	μ	Oil Viscosity	cP	0.8
M	M	Tangential Blocks		300
N	N	Radial Blocks		100
O	ϕ	Porosity		0.25
P	J	Streamlines per Block		2
Q	k_{ca}	Casing Permeability	mD	1×10^{-17}

To represent the productivity of a well two responses are recorded; flow rate and skin. The flow rate calculation is discussed in Section 3.8. As discussed in Section 1, skin is a means of measuring the difference in flow rate from that of the ideal flow conditions for a

homogenous isotropic reservoir. In Darcy's law for radial flow the skin term occurs in the denominator, as shown:

$$Q = \frac{2\pi Kh\left(P_e - P_{wf}\right)}{\mu\beta\left(ln\left(\frac{r_e}{r_w}\right) + S\right)}$$

Rearranging this for skin results in:

$$S = \frac{2\pi Kh\left(P_e - P_{wf}\right)}{\mu\beta Q} - ln\left(\frac{r_e}{r_w}\right)$$

In the above formula the calculated flow rate is used to determine the skin factor, providing an additional means to evaluate the effectiveness of the completion and the overall productivity. A summary of the responses for this case study is provided in Table 8.

Table 8 – Case Study 1: Responses

Response	Name	Units
1	Flow Rate	bbl/d
2	Total Skin	n/a

The experiment design selected for this case study is a 2^{6-1} fractional factorial design. This represents a half fraction two level design with six factors. A full factorial would require 64 runs. In running a half fraction only 32 runs are required. The half fraction results in a resolution VI design. This means single factors are aliased with five factor interactions, two factor interactions are aliased with four factor interactions and three factor interactions are aliased with each other. This is a high resolution design and results in clear estimates for main effects and two factor interactions.

A quarter fraction design could be run, representing a 2^{6-2} fractional factorial design. This would result in only 16 required runs. However, this is a resolution IV design in which two factor interactions would be aliased with other two factor interactions, resulting in difficulties in determining which interactions actually contribute to the experiment results. Therefore, a 2^{6-1} fractional factorial is determined to be the most appropriate design.

To test for the presence of curvature two additional center points are added. Normally only one center point is required. However, with the selection of a categoric variable, two are necessary; one at each level. This results in a total of 34 experiment runs. The run combinations and results are provided in Appendix II.

4.4.3 Statistical Analysis

A statistical summary of the output results is provided in Table 9. Note that over the provided ranges flow rate varied from 143 to 787 bbl/d and skin varied from -1.5 to 23. Recall that negative skin represents stimulation, while positive skin represents damage (as compare to the radial isotropic homogenous case).

The statistical analysis is broken down into two sections. In Section 4.4.3.1 the flow rate response is analyzed, while in Section 4.4.3.2 the skin response is analyzed.

Table 9 – Case Study 1: Statistical Summary

Name	Skin	Flow Rate
Units	n/a	bbl/d
Observations	34	34
Minimum	-1.49	142.6
Maximum	23.03	787.1
Mean	4.14	474.9
Standard Deviation	6.32	179.7

4.4.3.1 Flow Rate Response

In Figure 59 a half-normal probability plot is provided. In these plots unimportant effects fall on a straight line, while significant effects veer from it. The more significant the effect, the further it is from the straight line.

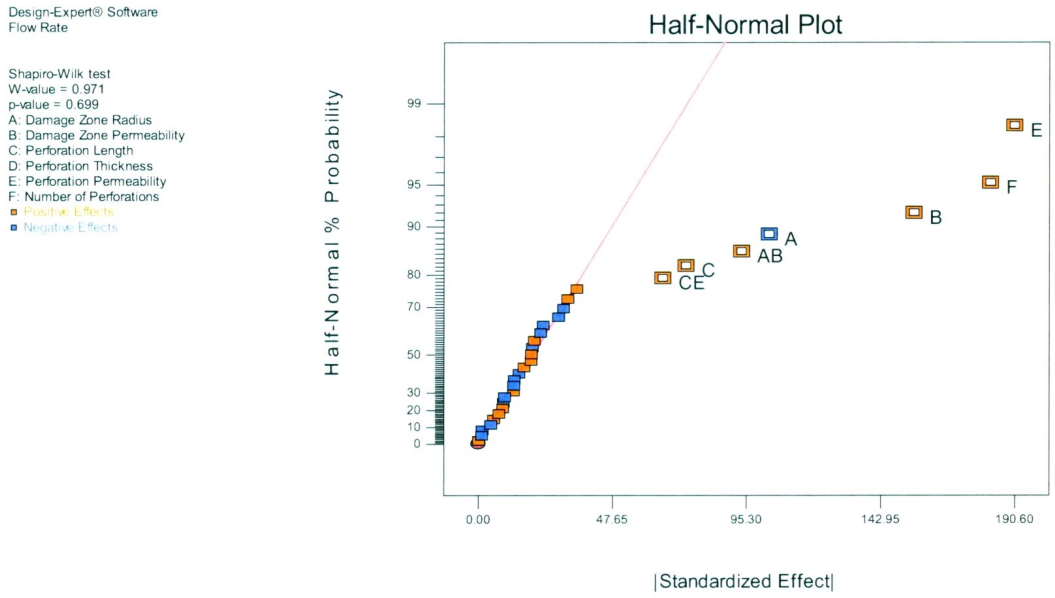


Figure 59 - Case Study 1: Flow Rate: Half Normal Probability Plot

The figure shows perforation permeability to be the most significant factor, followed closely by the number of perforations and the damage zone permeability. Other important effects include damage zone radius, perforation length and the interactions between damage zone radius and damage zone permeability as well as perforation length and perforation permeability. Note that the significance of the interactions could not have been determined without appropriate experiment design techniques.

Of the factors investigated, only perforation thickness proves to be statistically insignificant to the response. Physically, this is a realistic result. The thickness of the

perforation can only be varied over a limited range since the diameter of the shot required to make the perforation is limited. Over this range, the differences in flow rate are negligible. It is important to note that all observations made are dependent upon the range selected, referred to as the design space. Therefore, careful thought has been placed in selecting appropriate and realistic ranges for each factor.

The half-normal probability plot provides a visual representation of the important factors within the experiment. To provide a numerical measure of the statistical significance of each factor an ANOVA is performed. The results of the ANOVA are provided in Table 10. Note that only the significant factors are shown in the table.

Table 10 – Case Study 1: Flow Rate: ANOVA Results

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Model	9.818E+5	7	1.403E+5	53.63	< 0.0001
A - Damage Zone Radius	85700	1	85700	32.77	< 0.0001
B - Damage Zone Permeability	1.920E+5	1	1.920E+5	73.40	< 0.0001
C - Perforation Length	43688	1	43688	16.70	0.0004
E - Perforation Permeability	2.906E+5	1	2.906E+5	111.12	< 0.0001
F - Number of Perforations	2.456E+5	1	2.456E+5	93.90	< 0.0001
AB	70218	1	70218	26.85	< 0.0001
CE	34473	1	34473	13.18	0.0013
Curvature	21197	2	10598	4.05	0.0305
Residual	62769	24	2615		
Cor. Total	1.066E+6	33			

The ANOVA results confirm the previous observations. Only factor D, perforation thickness, is insignificant. Note that the table contains a row labeled curvature. The curvature of the model, measured at the center points, provides a P-value of 0.03. The significance of the term is dependent upon the chosen significance level for the experiment. In this case study a significance level of $\alpha = 1\%$ is chosen. The P-value is

greater than the significance level, meaning significant curvature exists within the model. To investigate this curvature a more complex experiment design is required. This is the subject of Case Study 2 (Section 4.5).

To confirm the results of the ANOVA general diagnostic tests are run to ensure the ANOVA assumptions have not been violated. These include normality, constant variance and randomness of the residuals. The results of these tests are provided in Figure 60.

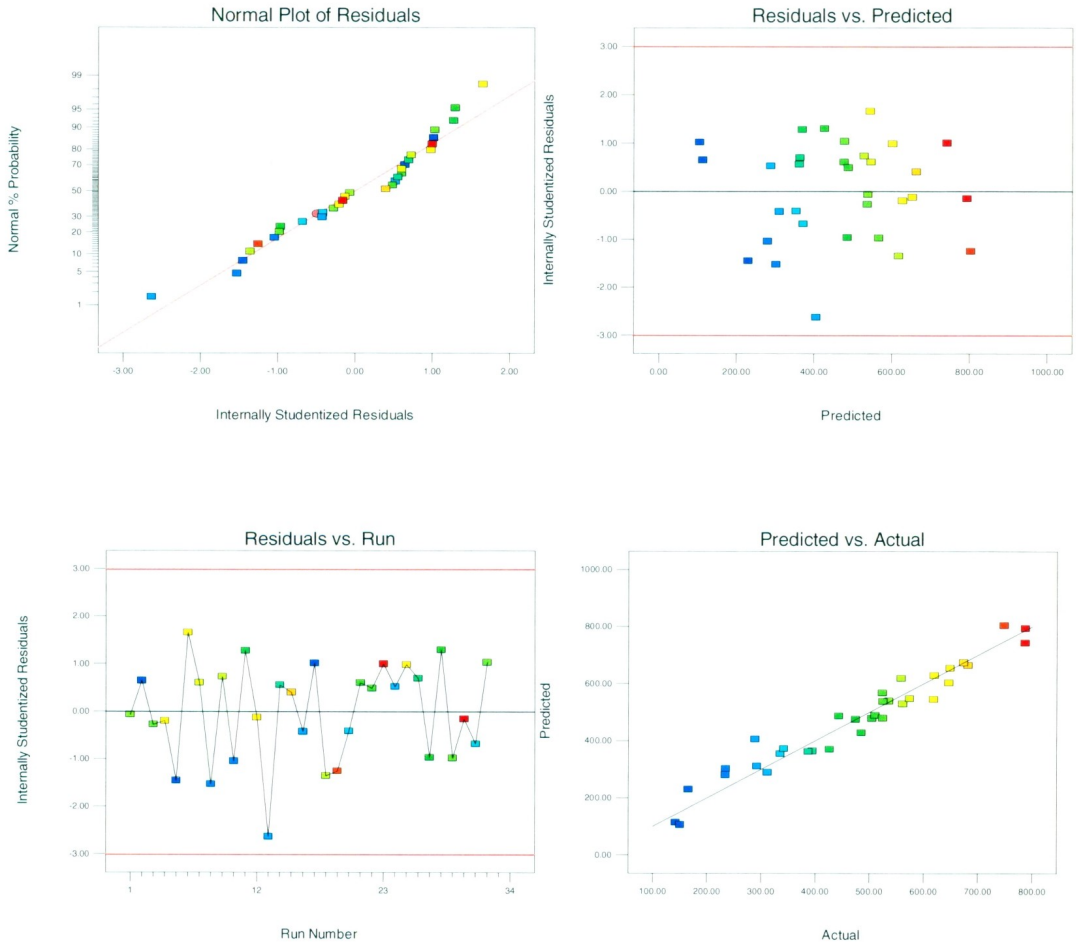


Figure 60 - Case Study 1: Flow Rate: ANOVA Assumptions Plots

The normal probability plot of the residuals shows the data fall on a straight line, meaning the residuals are distributed normally. The next graph portrays residuals versus the predicted value from the regression equation: randomly distributed data is observed; trends indicating that a different distribution should be used are not present. The residuals are fairly evenly distributed, proving the assumptions of normality and constant variance to be correct. The next graph portrays residuals versus run number. Trends indicating systemic errors are not apparent, as expected for numerical simulations. This verifies the randomness of residuals assumption is satisfied. Therefore, all assumptions are satisfied and the results can be trusted.

Note that this analysis is being used for factor screening only. The regression model generated is invalid due to the presence of curvature, therefore, the regression model and subsequent analysis is not discussed. Case Study 2 attempts to determine the true model behavior.

4.4.3.2 Skin Response

As with the flow rate response analyzed in Section 4.4.3.1, the same general factor screening analysis is performed for the skin response. However, for the skin response a natural log transform is required. This is determined by performing the general analysis and assessing the diagnostic plots for the ANOVA assumptions. The diagnostic plots are provided in Figure 61.

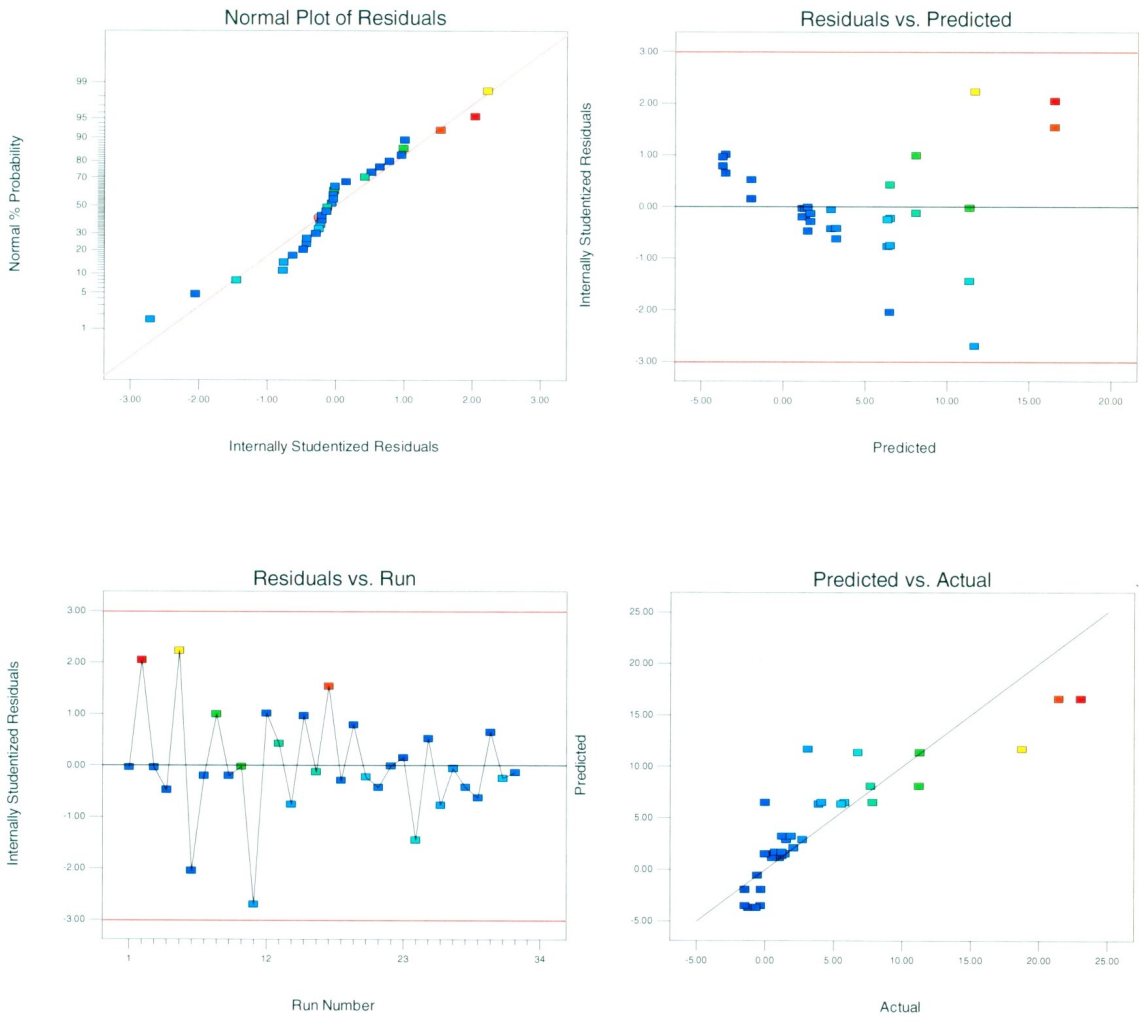


Figure 61 - Case Study 1: Skin: ANOVA Assumptions Plots

The normal probability plot shows a slight S-shaped trend, indicating the data to be non-normally distributed. The residual versus predicted value plot shows a coning trend in which the variance increases with predicted value. These observations conflict with the ANOVA assumptions of constant variance and normally distributed data, therefore, a transform is required.

Several options are available to determine the appropriate transform to apply. Within Design Expert™, a Box-Cox plot for power transforms indicates whether or not a

transform is required, as well as providing a suggestion for which transform to apply. The program suggests applying a square root transform. However, experience indicates that natural logarithm transformations are normally best when the ratio between the maximum and minimum response values is greater than 10. In this case, the ratio is difficult since the values of skin can be negative. Regardless, a maximum of 23 and a minimum of -1.5 is a strong suggestion that a natural logarithm transform may be required.

Since values of the skin response are negative, a straight forward natural logarithm cannot be performed. To fix this, all values are shifted by a constant to prevent negative values. Design Expert™ recommends a constant of $k = 1.63371$. Using this value, the data is transformed via $ln(Skin + k)$.

Performing the transform on the data, the general analysis can be carried out. The half-normal probability plot is provided in Figure 62.

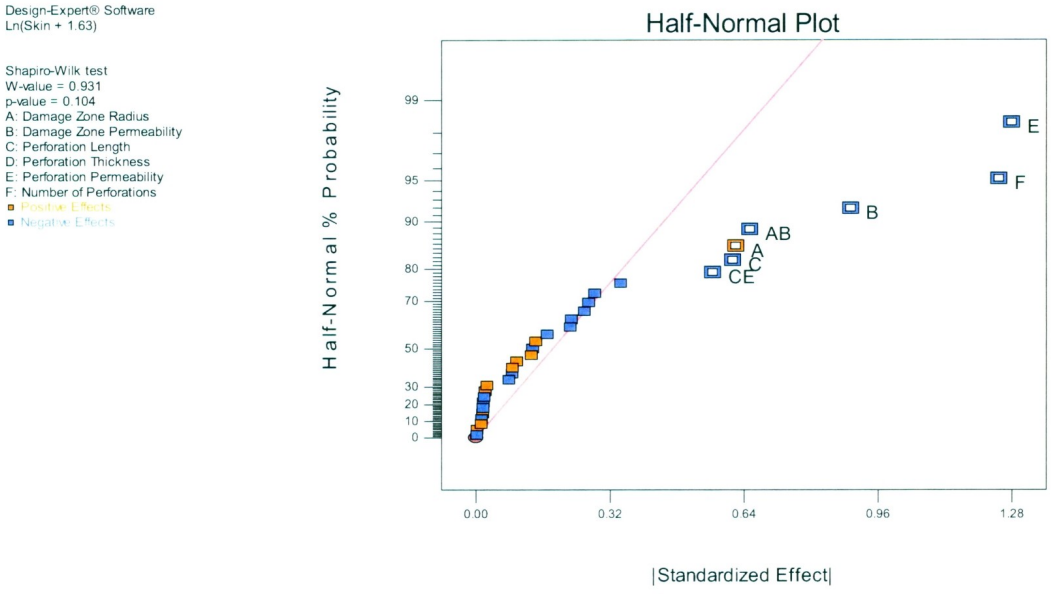


Figure 62 - Case Study 1: ln(Skin + k): Half Normal Probability Plot

With the transformed data the results indicate the same significant factors and interactions as in the flow rate response. The only insignificant factor is the perforation thickness. All other factors, as well as the two interactions, are significant to the results.

The remainder of the ANOVA analysis has been performed but is not provided due to its redundancy. All parameters in the ANOVA match similarly to the ANOVA results of the flow rate response. As well, all ANOVA assumptions are valid with the transformation of the data.

4.4.4 Conclusions

In this case study a 2^{6-1} fractional factorial design was used as a means of factor screening in a cased and perforated well completion within a near wellbore streamline model. A visual of the typical streamlines distribution in these cases is provided in Figure 63.

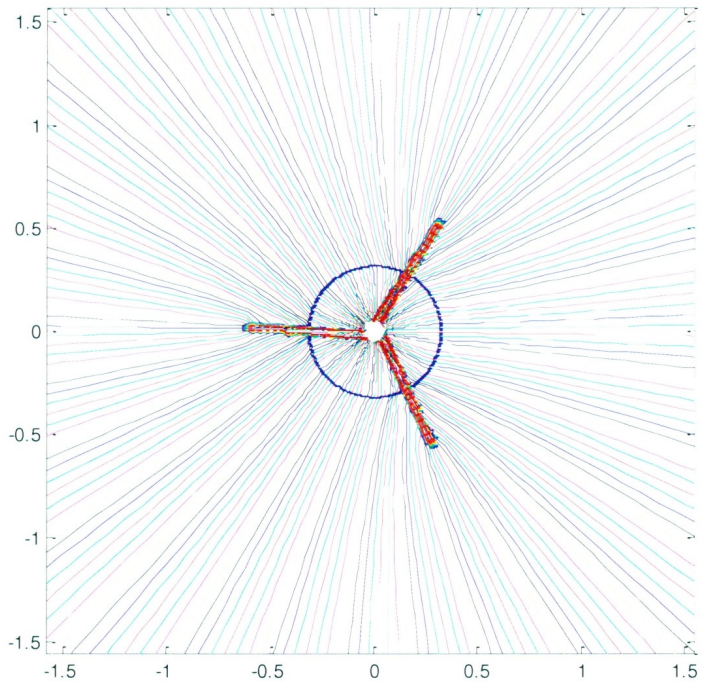


Figure 63 - Cased and Perforated Streamline Distribution

The figure shows how the perforations and near wellbore damage influence the streamlines throughout the near wellbore region.

This case study has determined the significant factors influencing the productivity of a well, including the number of perforations, perforation permeability, perforation length, damage zone radius and damage zone permeability. In addition, two interactions are also significant: the interaction between damage zone radius and damage zone permeability; and the interaction between perforation length and perforation permeability. This information was achieved through 34 runs in a resolution VI half fraction factorial design. The experiment also showed the presence of significant curvature within the model, indicating the need for a more complex experiment design to capture the true nature of the model behavior. This is the subject of the next case study.

4.5 Case Study 2: Three Factor Cased and Perforated Well

4.5.1 Study Parameters and Assumptions

In Case Study 1 a 2^{6-1} fractional factorial design was used to determine the significant effects which influence the productivity of a cased and perforated well. Five factors were deemed statistically significant, including the number of perforations, perforation permeability, perforation length, damage zone permeability and damage zone radius. Two interactions were also deemed significant. The experiment revealed significant curvature within the results, highlighting the need for a more complex design to fully capture the behaviour within the model.

The purpose of this case study is to build upon the lessons learned in Case Study 1. The goal is to use a superior experiment design capable of modeling the curvature within the model. By accurately matching a response surface to the model, the behaviour of its response with respect to the variable inputs can be captured. This will show which factors affect productivity, how they affect the productivity and which combinations lead to the most effective completion.

The factors under study within this case study are altered from those used in Case Study 1. As oppose to continuing with five significant factors, only three are investigated. The factors to be studied include the number of perforations, perforation length and perforation permeability. There are three main reasons for this change.

First, the neglected variables (damage zone radius and damage zone permeability) have been well studied within the available literature. The Hawkin's formula accurately represents their impact on flow, and is widely accepted to be an accurate representation

of reality. In addition, runs of the near wellbore streamline model with only a damage zone present leads to results which accurately match those predicted by the Hawkin's formula.

Second, within the industry, once a well has been drilled and casing placed, little can be done to alter the near wellbore damage associated with drilling. Neglecting forms of stimulation treatments, the only variables that can be altered at this point are the perforation characteristics, therefore, the focus of this case study is solely on the perforations themselves. The near wellbore damage parameters still exist within the model, but remain constant throughout the study. Therefore, this is an adequate and realistic assumption.

Lastly, in reducing the number of parameters under study, the number of runs required in the experiment design is reduced. Since higher order models are required to capture the nature of the curvature within the model, more runs are required. Reducing the number of variables will consequently result in less runs, regardless of the order of the model. Therefore, this case study focuses solely on the perforation characteristics in a cased and perforated well, investigating their effect on productivity. To do this, an IV-optimal design is performed. Detail on IV-optimal designs is provided in Section 4.2.4.2.

4.5.2 IV-Optimal Design Setup

As discussed, in this case study three factors are selected to investigate their influence on the productivity of a well. The factors, along with their corresponding levels or ranges, are provided in Table 11.

Table 11 – Case Study 2: Factor Summary

Factor	Name	Units	Low (Actual)	Medium (Actual)	High (Actual)
A	Perforation Length	m	0.15	n/a	1.0
B	Perforation Permeability	mD	100	n/a	1600
C	Number of Perforations	n/a	1	2	3

The table provides high and low levels for perforation length and perforation permeability. These are numeric factors. The IV-optimal design uses these values as ranges for the design space. Using an algorithm, points are generated throughout the design space to capture the nature of the response. The number of points is dependent upon the order of the model selected.

The last factor under study, number of perforations, is selected as a categoric variable with three levels. In selecting this to be a categoric variable no values are generated at unfeasible values in between the levels. Therefore points for the number of perforations may only be run at 1, 2 or 3. The latest version of Design Expert™ allows for the selection of discrete variables. This presents a new technique to deal with this circumstance, but is not performed in this case study.

Numerous other factors play a role in influencing productivity. The majority of the constants remain the same as Case Study 1, with values provided in Table 7. The new constants for this case study are provided in Table 12.

Table 12 – Case Study 2: Additional Constants

Factor	Symbol	Name	Units	Value
D	r_d	Damage Zone Radius	M	0.5
E	k_d	Damage Zone Permeability	mD	50
F	t_p	Perforation Thickness	M	0.04

As with the previous case study, two responses are recorded; flow rate (in bbl/d) and total skin. Both are good representatives of productivity. An effective completion will maximize flow rate and minimize skin.

To model the curvature known to exist within the model, an IV-optimal design is utilized. Prior to performing the experiment the degree of curvature is unknown. Within Design Expert™ IV-optimal designs have the ability to model up to sixth order polynomial behavior. This is also a sequential design in which additional runs may be performed in a successive nature to increase the order of the desired response surface.

The near wellbore streamline model under study is extremely fast in its calculation process, providing results for nearly 100 000 grid blocks in a matter of minutes on a standard 64 bit windows system with 4 GB of RAM. Therefore, a large number of runs are not an issue with this experiment.

As opposed to augmenting the design for more runs when more complex curvature is observed, this experiment is set up to handle a quartic response from the beginning, resulting in a total of 40 required runs. As will be discussed throughout the next section, a quartic polynomial is deemed appropriate for the skin response. To ensure a higher order model is not necessary, the design is augmented to allow for the matching of a fifth order polynomial. This results in an additional 22 runs, leading to a total of 62. The run combinations and results are provided in Appendix III. The additional points from augmentation also serve as verification runs, verifying the response surface results.

4.5.3 Statistical Analysis

As with Case Study 1, the statistical analysis is broken down into two sections. In Section 4.5.3.1 the flow rate response is analyzed, while in Section 4.5.3.2 the skin response is analyzed.

4.5.3.1 Flow Rate Response

The first part of the analysis involves determining the order of the model deemed appropriate to match the response. A fit summary of each available model type is provided in Table 13. Two coefficients of determination (R-squared) values are provided to determine the adequacy of the match.

Table 13 – Case Study 2: Flow Rate: Fit Summary

Model	Sequential P-value	Adjusted R-Squared	Predicted R-Squared	Program Recommendation
Linear	< 0.0001	0.8691	0.8538	
2FI	0.0002	0.9107	0.8886	
Quadratic	< 0.0001	0.9808	0.9750	
Cubic	< 0.0001	0.9972	0.9954	
Quartic	< 0.0001	0.9993	0.9985	Suggested
Fifth	0.00120	0.9998	0.9982	Suggested
Sixth				Aliased

The table shows R-squared values greater than 0.99 for cubic models and above. Design Expert™ suggest using either a quartic or fifth order polynomial model. However, the suggestion is not always the best solution and may lead to over complication within the model. The cubic model provides very similar R-squared values, and further analysis proves it to be adequate. Therefore, a cubic response surface is used to match the flow rate response.

The next step of the analysis involves performing an ANOVA to determine which factors, corresponding interactions and higher order terms contribute to the response. Several computerized options are available to determine the required parameters, including forward elimination, backward elimination and stepwise elimination at a chosen significance level. As oppose to the computerized process, a manual selection process is performed, allowing for greater control and insight into the selection process. Focus is placed on both the F-value and P-value for each model term, eliminating the terms for F-values less than 10. The results of the ANOVA are provided in Table 14. Note that only the significant factors are shown in the table.

Table 14 – Case Study 2: Flow Rate: ANOVA Results (Cubic Model)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Model	7.452E+5	9	82802	631	< 0.0001
A - Perforation Length	63889	1	63889	487	< 0.0001
B - Perforation Permeability	2.695E5	1	2.695E+5	2055	< 0.0001
C - Number of Perforations	3.773E5	2	1.887E+5	1439	< 0.0001
AB	24758	1	24758	188	< 0.0001
A^2	11339	1	11339	86	< 0.0001
B^2	35066	1	35066	267	< 0.0001
A^2B	3544	1	3544	27	< 0.0001
B^3	5061	1	5061	38	< 0.0001
Residual	6816	52	131.1		
Cor Total	7.689E+5	61			

The ANOVA reveals all three main factors are significant to the flow rate response. In addition, the interaction between perforation length and perforation permeability is statistically significant, confirming the results from Case Study 1. Higher order terms are also significant, up to a third order term for perforation permeability. The model has an R-squared of 0.9909, an adjusted R-squared of 0.9894 and a predicted R-squared of 0.9868. These values are high and are in reasonable agreement, indicating a valid model.

Prior to investigating the nature of the model results, the underlying assumptions within the ANOVA must be checked. These include normality, constant variance and randomness of the residuals. General diagnostic tests are run to ensure the ANOVA assumptions have not been violated. The results of these tests are provided in Figure 64.

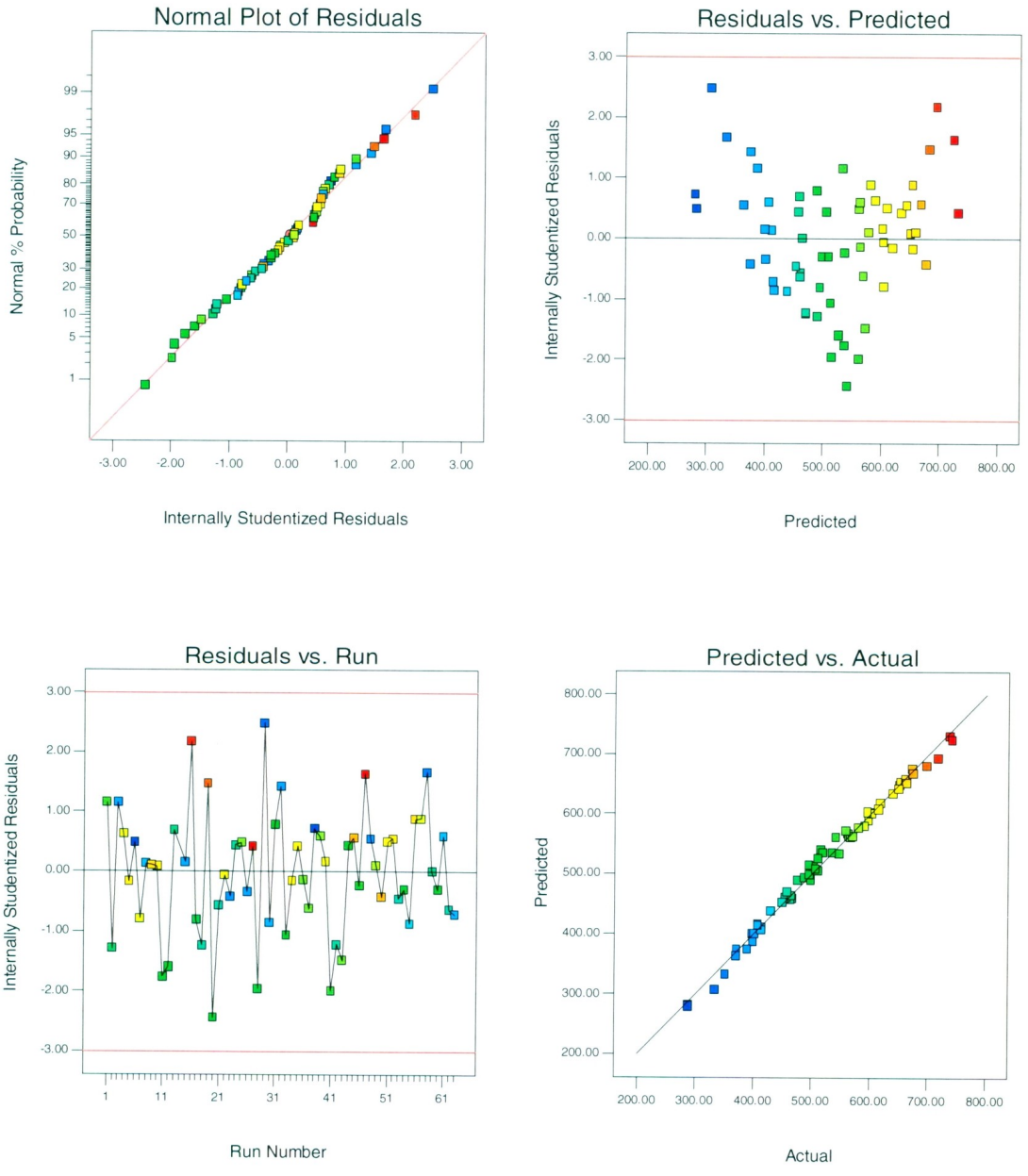


Figure 64 - Case Study 2: Flow Rate: ANOVA Assumptions Plots

The results of the diagnostic plots validate that the ANOVA assumptions have not been violated. In the normal probability plot all points are located around the straight line, indicating a normal distribution. In the residual versus predicted value plot the data has constant variance within the defined limits. In the residuals versus run number plot, no systematic trends are observed, indicating randomness. Finally, in the predicted versus actual value plot, a good match is observed between the experiment results and the predicted values of the model. Therefore, the results of the ANOVA and the resultant model can be trusted to accurately represent the experiment results.

The goal of this case study is to understand the model behavior, however, since the response surface is created, the meta-model can be shown. It is important to remember that while the meta-model represents the results of this case study, it does not represent the model as a whole. If any of the parameters are changed, including the numerous constants assumed throughout, then the meta-model will not be valid. Also, even given the same conditions, the model will only be applicable within the design space. Therefore, it is acceptable for local analysis only, and will not provide accurate results outside of the ranges for each factor.

With this in mind, the meta-model will take the following form:

$$Q = \mathrm{X}_1 + \mathrm{X}_2 L_p + \mathrm{X}_3 K_p + \mathrm{X}_4 L_p K_p + \mathrm{X}_5 {L_p}^2 + \mathrm{X}_6 {K_p}^2 + \mathrm{X}_7 {L_p}^2 K_p + \mathrm{X}_8 {K_p}^3$$

Here, L_p is the perforation length while K_p is the perforation permeability. Note that the equation does not contain the number of perforations. Since this is a categoric variable, three separate equations are required; one for each level within the number of

perforations (1, 2 or 3). As well, for the equation to be valid the units must equal the units of the input variables. These are:

$$L_p = [m], K_p = [mD], Q = [bbl/d]$$

The values of the coefficients for each level within the number of perforations are provided in Table 15.

Table 15 – Case Study 2: Flow Rate: Cubic Model Equation Coefficients

Value of Coefficients	1 Perforation	2 Perforations	3 Perforations
Intercept	220.24	354.72	408.18
L_p	39.21	39.21	39.21
K_p	0.3923	0.3923	0.3923
$L_p K_p$	0.3402	0.3402	0.3402
${L_p}^2$	-29.47	-29.47	-29.47
${K_p}^2$	-3.968×10^{-4}	-3.968×10^{-4}	-3.968×10^{-4}
${L_p}^2 K_p$	-0.1954	-0.1954	-0.1954
${K_p}^3$	1.1303×10^{-7}	1.1303×10^{-7}	1.1303×10^{-7}

Note that since the number of perforations (term C) was found to be a linear factor, the only change in the coefficients relates to the intercept. The remaining terms are unchanged.

Trusting the validity of the model, the model behavior may be analyzed. Figure 65 provides contour plots of the flow rate response.

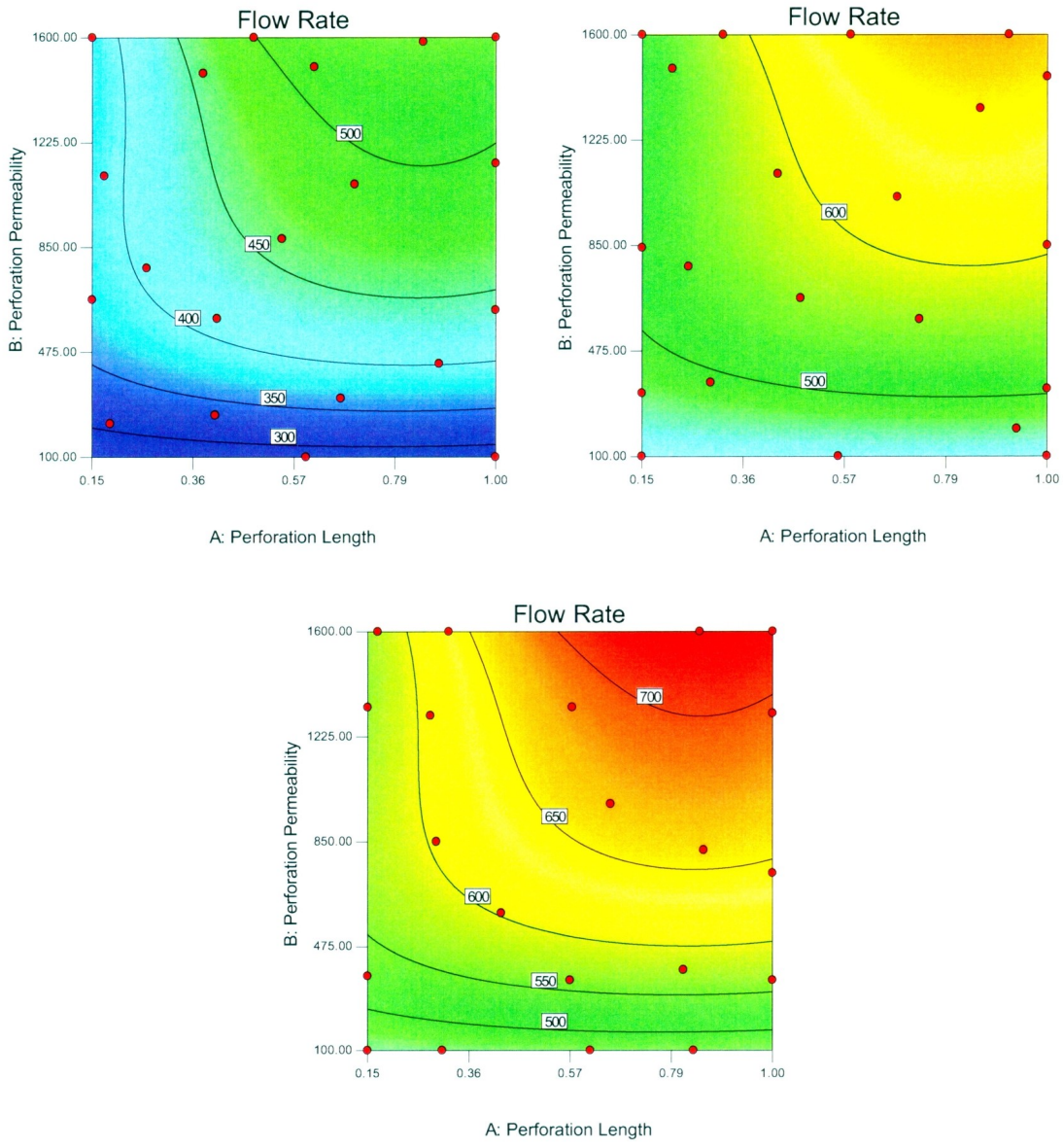


Figure 65 - Case Study 2: Flow Rate: Contour Plots (1, 2 and 3 Perforations)

Note that the three subfigures are shown for 1 perforation, 2 perorations and 3 perforations, respectively. To further enhance the clarity of the model response three dimensional surface plots are provided. Two viewpoints are provided for each level of number of perforations. The rows are for 1, 2 and 3 perforations, respectively.

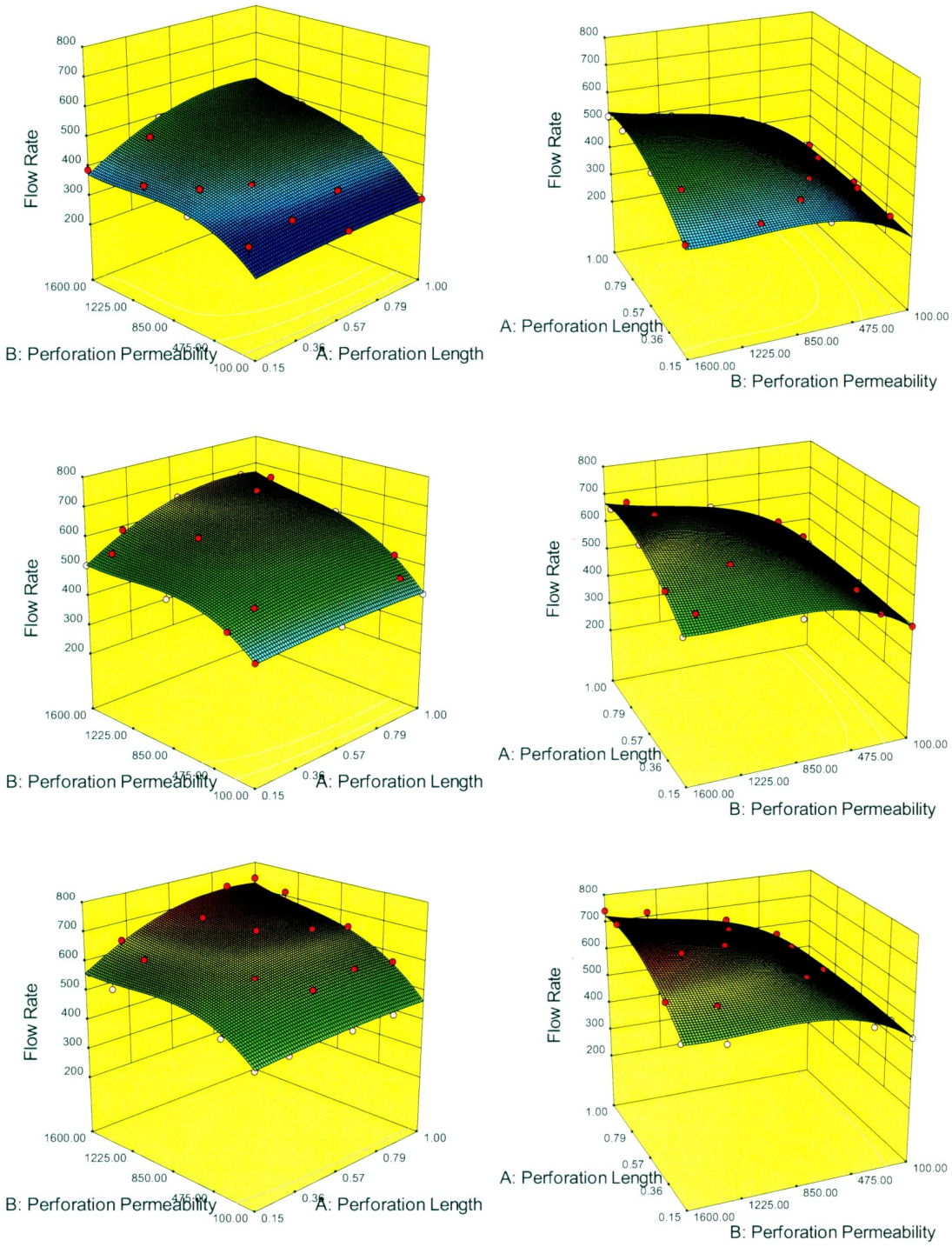


Figure 66 - Case Study 2: Flow Rate: 3D Surface Plots

An analysis of both the contour plots and response surfaces provides valuable insight into the model behaviour. It is clear that for ideal perforation effectiveness all factors should

be at their maximum levels. Beyond this, the behaviour of the individual factors reveals interesting trends. For the number of perforations, the increase in productivity follows a linear trend independent of the other variables. The remaining variables have more complex behaviour. Perforation length has a quadratic response, perforation thickness has a cubic response, and strong interactions occur between the two variables. For instance, when perforation permeability is at its minimum, there is little to no increase in flow rate as a result of increasing the perforation length. However, as the perforation permeability increases, the effect of perforation length also increases, becoming a more important factor in the completion effectiveness. This shows that while each variable in itself is important to the productivity of a well, it is the interaction between them that must be focused upon to maximize completion effectiveness.

4.5.3.2 Skin Response

The skin response is analyzed in the same manner as the flow rate response in the previous section. Table 16 provides a fit summary of each available model type. The sequential P-value and two coefficients of determination are provided to assess the adequacy of the match.

Table 16 – Case Study 2: Skin: Fit Summary

Model	Sequential P-value	Adjusted R-Squared	Predicted R-Squared	Program Recommendation
Linear	< 0.0001	0.8012	0.7746	
2FI	< 0.0001	0.9443	0.9280	
Quadratic	< 0.0001	0.9615	0.9440	
Cubic	< 0.0001	0.9951	0.9904	
Quartic	< 0.0001	0.9988	0.9961	Suggested
Fifth	0.0038	0.9996	0.9809	Suggested
Sixth	< 0.0001	1.0000		Aliased

The table shows a quartic model and above have R-squared values greater than 0.99. Design Expert™ recommends a quartic or fifth order response. Careful analysis, as described in the remainder of this section, reveals a quartic response surface to be the appropriate choice. A lower order cubic model can be utilized, but significant information relating to the complexity of the response may be lost in utilizing this selection. Therefore, the quartic response is the subject of this analysis.

To determine the significant parameters affecting the skin response an ANOVA is performed. As with the flow rate response, a manual selection process is utilized to determine the significance of each model term. Focus is placed on the P-values and F-values, eliminating terms with F-values less than 10. The results of the ANOVA, showing the significant model terms, are provided in Table 17.

Table 17 – Case Study 2: Skin: ANOVA Results (Quartic Model)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Model	251.0	13	19.3	310	< 0.0001
A - Perforation Length	12.3	1	12.3	198	< 0.0001
B - Perforation Permeability	87.0	1	87.0	1399	< 0.0001
C - Number of Perforations	127.5	2	63.8	1026	< 0.0001
AB	4.35	1	4.35	70	< 0.0001
BC	7.07	2	7.07	57	< 0.0001
A^2	3.68	1	3.68	59	< 0.0001
B^2	19.4	1	19.4	312	< 0.0001
B^2C	1.88	2	0.94	15	< 0.0001
B^3	3.94	1	3.94	66	< 0.0001
B^4	1.22	1	1.22	19	< 0.0001
Residual	2.98	48	0.062		
Cor Total	254.0	61			

As with the flow rate response and the results of Case Study 1, all three main effects and the interaction between perforation length and perforation permeability are significant to

the skin response. However, more complex behavior is observed. With respect to perforation length, a quadratic response is observed. Perforation permeability also shows more complex behavior, going up to a quartic response as well as having several interaction effects. In addition, the number of perforations is also showing more complex behavior; the main response remains linear, but several additional interactions exist which were not observed in the flow rate response.

The model created has an R-squared of 0.9883, an adjusted R-squared of 0.9851 and a predicted R-squared of 0.9784. As with the flow rate model, the values are high and are in reasonable agreement, indicating a valid model.

Prior to reviewing and investigating the nature of the model results for the skin response, the underlying assumptions within the ANOVA must be checked. The results of the general diagnostic tests are provided in Figure 67.

The results of the diagnostic plots confirm the ANOVA assumptions have not been violated. In the normal probability plot the majority of the points are located around the straight line, indicating a normal distribution. Slight variation is observed, but it is within an acceptable standard with a negligible S-shaped occurrence. The majority of the points in the residual versus predicted value plot have a constant variance within the defined limits. One outlier exists on the fringes of the line, but can be accepted at this level. A slight pattern of increasing variance with increasing predicted value can be observed. However, applying a natural log transform over compensates for this, leading to an inverse coning trend. Therefore the normal distribution is chosen to be the appropriate distribution for the data.

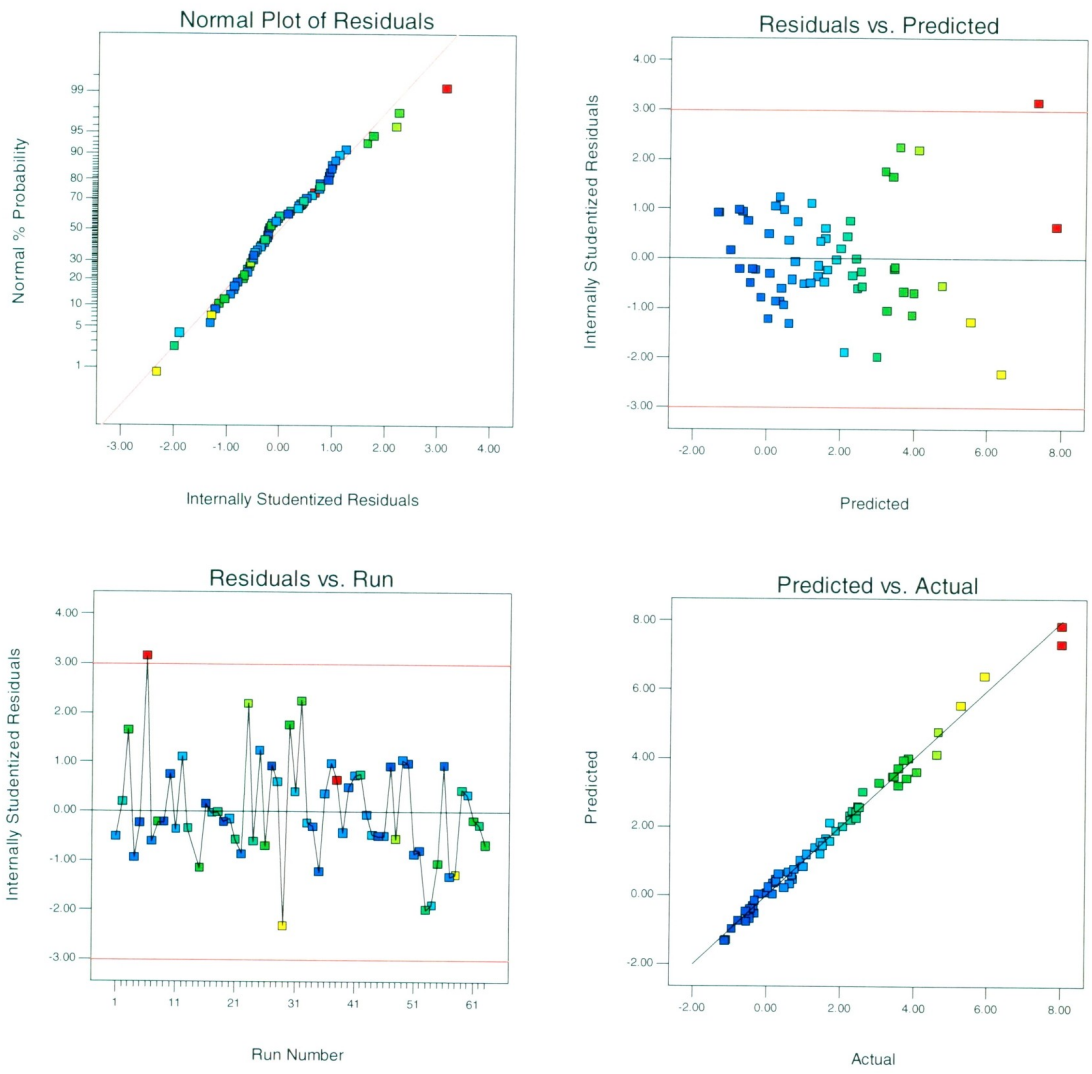


Figure 67 - Case Study 2: Skin: ANOVA Assumptions Plots

In the residuals versus run number plot, no systematic trends are observed, indicating randomness. Finally, in the predicted versus actual value plot, a good match is observed. Therefore, the results of the ANOVA and the resultant model can be trusted to accurately represent the experiment results.

While the mathematical meta-model is not the focus of this case study, it is provided to enhance the understanding of the behavior within the model. As with the flow rate meta-

model, the meta-model provided is only useful for local analysis, and will not be valid outside of the factor ranges, or if any of the numerous constants are altered.

The meta-model for the skin response will take the following form:

$$Skin = \mathrm{X}_1 + \mathrm{X}_2 L_p + \mathrm{X}_3 K_p + \mathrm{X}_4 L_p K_p + \mathrm{X}_5 {L_p}^2 + \mathrm{X}_6 {K_p}^2 + \mathrm{X}_7 {K_p}^3 + \mathrm{X}_8 {K_p}^4$$

Again, L_p is the perforation length and K_p is the perforation permeability. The equation does not contain the number of perforations since it is a categoric variable. Therefore, three equations are necessary; one for each level of the number of perforations. As well, for the equations to be valid the units must equal the units of the input variables. These are $L_p = [m]$ and $K_p = [mD]$. The values of the coefficients for each level within the number of perforations are provided in Table 18.

Table 18 – Case Study 2: Skin: Quartic Model Equation Coefficients

Value of Coefficients	1 Perforation	2 Perforations	3 Perforations
Intercept	10.44	5.92	4.91
L_p	-4.12	-4.12	-4.12
K_p	-0.0206	-0.0166	-0.0159
$L_p K_p$	-1.546×10^{-3}	-1.546×10^{-3}	-1.546×10^{-3}
${L_p}^2$	3.47	3.47	3.47
${K_p}^2$	2.859×10^{-5}	2.697×10^{-5}	2.674×10^{-5}
${K_p}^3$	-1.851×10^{-8}	-1.851×10^{-8}	-1.851×10^{-8}
${K_p}^4$	4.533×10^{-12}	4.533×10^{-12}	4.533×10^{-12}

Very little change is observed for the coefficients spanning the number of perforations. As with the results of the flow rate response, the number of perforations has a mostly linear effect. This forces changes within the intercepts for the equations, but does not affect the other higher order model terms. Significant interactions were observed between

the number of perforations and the perforation permeability in the ANOVA results. However, their effect cannot be seen in the coefficients above.

With the ANOVA assumptions verified the model may be analyzed. Figure 68 provides contour plots showing the skin response, while Figure 69 provides three dimensional surface plots to further enhance the clarity of the model. Note that for each figure three subfigures are provided. These are for 1, 2 and 3 perforations, respectively.

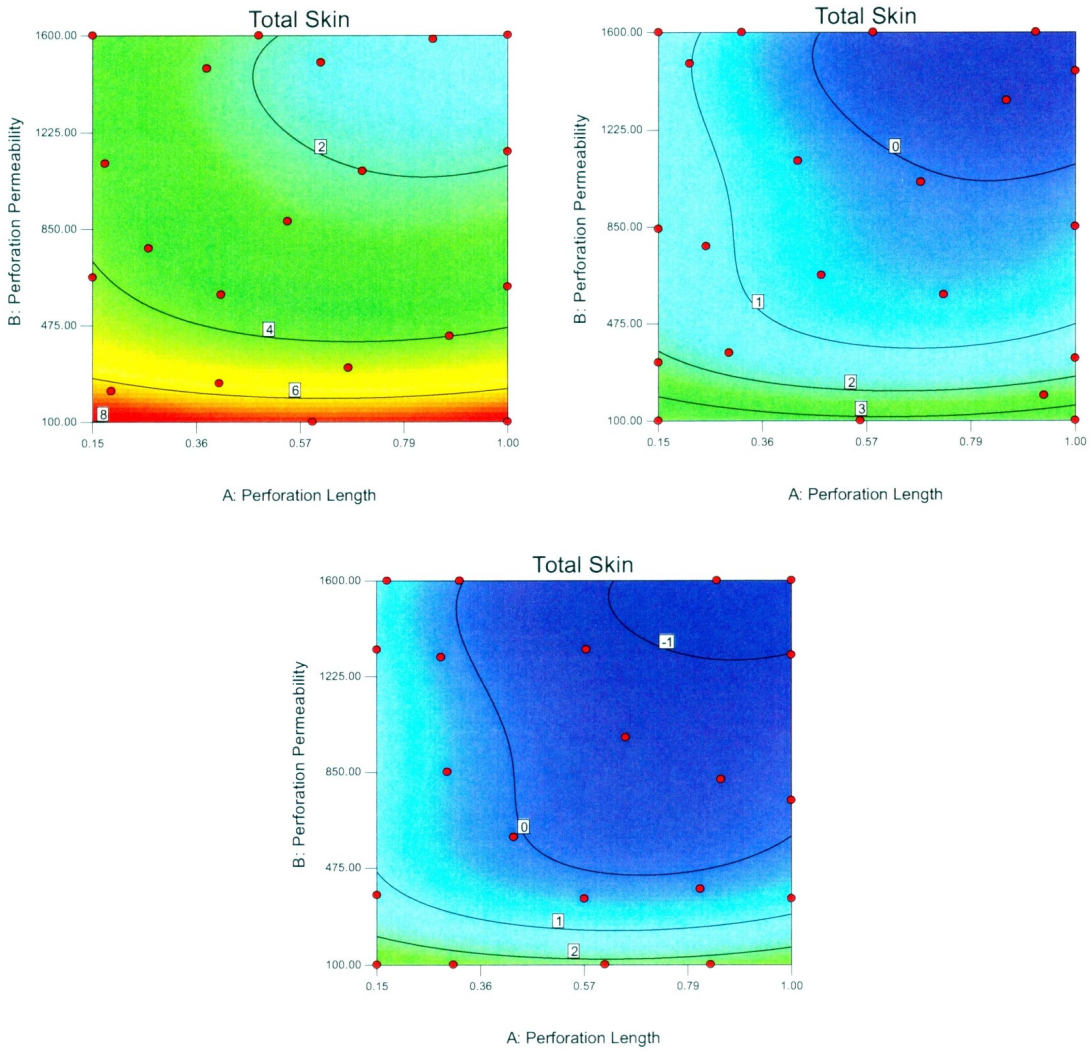


Figure 68 - Case Study 2: Skin: Contour Plots (1, 2 and 3 Perforations)

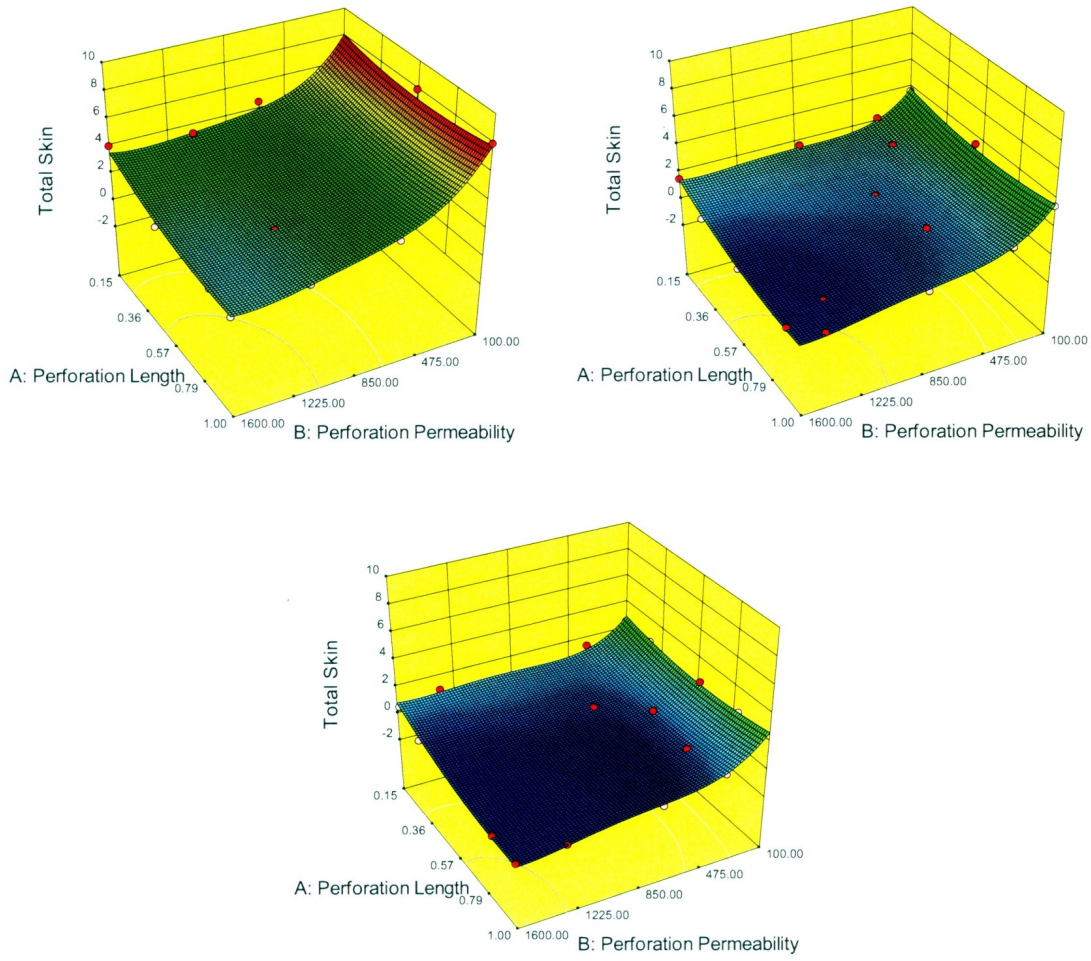


Figure 69 - Case Study 2: Skin: 3D Surface Plots

The plots indicate the complex model behavior. As observed, the optimal completion corresponding to minimum skin occurs when perforation length, perforation permeability and the number of perforations are all at their maximum values. Each factor within the model follows complex trends. The number of perforations follows a predominantly linear trend, the perforation length follows a quadratic trend, while the perforation permeability follows a quartic trend. Numerous interactions are also present.

The strong interactions between perforation length and perforation permeability are apparent within the contour plots, as the contour level increases from the bottom left to the top right. The understanding of this interaction is essential. As can be seen, at low levels of perforation permeability, changes in the perforation length do not significantly affect the skin response. However, as the perforation permeability is increased, changes in perforation length become much more significant, with greater length leading to less skin.

From the plots provided it is apparent that to achieve optimal completion effectiveness all three variables must be considered. Obtaining optimal perforation permeability must be the focus of any completion operation. If this is not achieved, the perforation will be below optimum, with extended perforation length not significantly enhancing flow. However, with high perforation permeability an optimal completion may be achieved, with efforts to extend the perforation length leading to a substantial decrease in skin. In addition, increasing the number of perforations (assuming them to be exact replicas and equally spaced) will result in a significant decrease in skin, thereby enhancing production.

The next section summarizes the results of this case study, as well as highlighting the changes in the streamline distributions.

4.5.4 Conclusions

In this case study an IV-optimal design was used to determine the nature of various factor influences in a cased and perforated well completion within a near wellbore streamline model. All three factors under study proved significant to the model behavior. The

precise nature of the influence of each factor is slightly dependent upon the response analyzed. For flow rate, the number of perforations results in linear behavior, perforation length results in quadratic behavior and perforation permeability results in cubic behavior. Strong interactions are also observed. For skin, a slight increase in model complexity is required. While the behavior of the number of perforations remains linear and perforation length remains quadratic in nature, the perforation permeability requires quartic behavior. Several additional interactions are also significant within the skin response.

The most significant conclusion that can be drawn from this analysis is related to the strong interaction between perforation permeability and perforation length. When perforation permeability is low, changes in perforation length do not result in increased productivity. However, as perforation permeability increases, the effect of perforation length becomes more substantial. Therefore, to provide maximum completion effectiveness high perforation permeability must be targeted as the main factor to optimize. In turn, this will lead to greater productivity as a result of perforation length, and thereby provide optimal completion performance.

Throughout this analysis the effect of various parameters has been determined, but little focus has been given to the nature of the streamline distributions. Figure 70 shows the effect of the number of perforations on the streamline distribution within the near wellbore model.

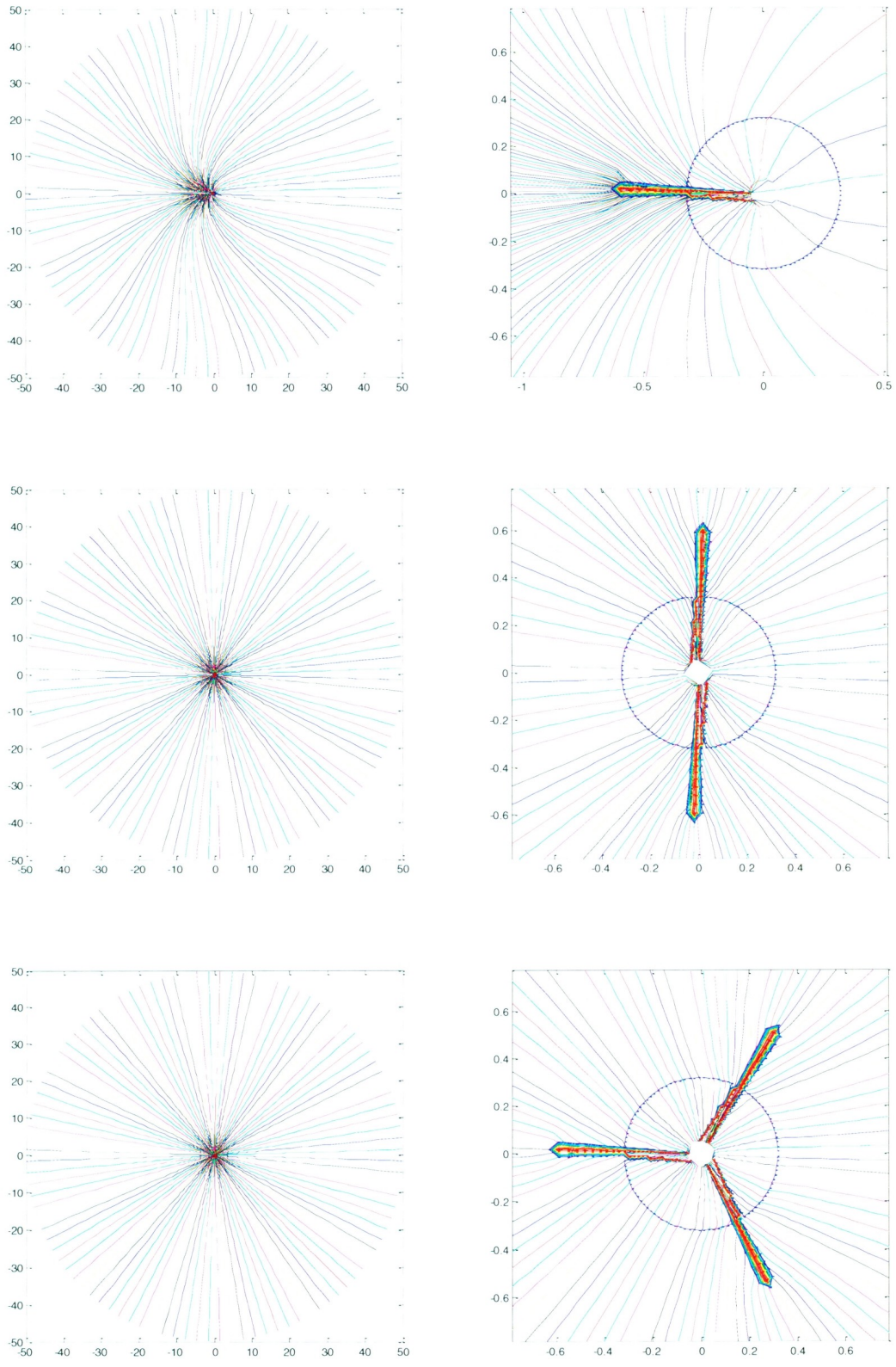


Figure 70 - Influence of Number of Perforations on Streamline Distribution

In the figure significant variations in curvature can be seen as a result of the different completions. Streamlines begin to flow radially from the external boundary of the reservoir in the direction of the wellbore. As the distance to the wellbore diminishes, the effect of the completion on the pressure distribution becomes more significant. This forces the streamlines to bend, flowing in the direction of the highest pressure differential. This phenomena forces streamlines to be attracted to areas of high permeability, forcing them towards and into the perforations.

Throughout this case study the effect of the number of perforations has proven to be a predominantly linear trend. As the number of perforations increase, the flow rate increases and skin decreases. This result can be related to the curvature shown within the figures. For one perforation significant curvature must occur in many streamlines for them to make their way through the reservoir and into the perforation. For a streamline to curve a higher pressure gradient is required. The loss in pressure results in less productivity. Therefore, the observation of significant curvature directly relates to a reduction in well productivity. As the number of perforations is increased, the curvature within the streamlines is reduced. Therefore, it is apparent that the more perforations present, the greater the productivity of a well.

In summary, for a cased and perforated well completion focus must be placed on all three factors. Maximum levels of each result in optimal well productivity. However, significant focus must be placed on maximizing perforation permeability due to its strong interaction with perforation length. In optimizing this parameter and maximizing the remaining parameters to reasonable limits, maximum completion effectiveness will be achieved.

4.6 Case Study 3: Filter Cake Buildup in Open-hole Completions

4.6.1 Study Parameters and Assumptions

The following case study deals with determining the effect of filter cake buildup in an open-hole completion. A description of the completion type is provided in the Section 4.3, along with a description of how the completion is represented in the near wellbore streamline model.

The purpose of this case study is to determine which factors pertaining to filter cake affect the productivity of a well, how they affect the productivity, and determine the most effective strategy to enhance the effectiveness of the completion. To do this, an IV-optimal design is utilized. As with Case Study 2, this design is chosen to best represent the expected complexity of the response within the numerical simulation.

4.6.2 IV-Optimal Design Setup

In this case study the properties of filter cake are investigated to determine their impact on the productivity of an open-hole completion. Three factors are used to represent the properties of the filter cake. The factors, along with their corresponding ranges, are provided in Table 19. Note that all three factors are numeric in nature.

Table 19 – Case Study 3: Factor Summary

Factor	Name	Units	Low (Actual)	High (Actual)
A	Filter Cake Permeability	μD	10	200
B	Filter Cake Thickness	mm	1	9
C	Angle of Coverage	deg	45	270

The table provides high and low values for filter cake permeability, filter cake thickness and angle of coverage. Note that the filter cake permeability is very low, in the micro-Darcy range. In addition, the filter cake thickness is very thin, in the millimeter range. These values are realistic and have been derived from the works of Pitoni & Ballard (1999) and Ochi et al. (1999). The angle of coverage is chosen to represent the area of the internal face covered by filter cake. This choice assumes that the filter cake covers the entire height of the reservoir layer within the model, over the extent of the angle of coverage within the wellbore.

Numerous other factors play a role in influencing productivity. Variables that are not under study within this case study are left as constants. The constants for this case study are provided in Table 20.

Table 20 – Case Study 3: Constants

Factor	Symbol	Name	Units	Value
D	k	Bulk Permeability	mD	100
E	k_{fs}	Free Space Permeability	D	100
F	r_e	External Radius	m	50
G	r_w	Wellbore Radius	m	0.0608
H	r_{if}	Internal Face Radius	m	0.05
I	p_e	External Pressure	bar	300
J	p_{if}	Internal Face Pressure	bar	280
K	μ	Oil Viscosity	cP	0.8
L	M	Tangential Blocks		300
M	N	Radial Blocks		300
N	ϕ	Porosity		0.25
O	J	Streamlines per Block		2

As with the previous two case studies, two responses are recorded; flow rate (in bbl/d) and total skin. Both are good representatives of the productivity of a well.

To model the curvature expected within the model an IV-optimal design is utilized. Information concerning IV-optimal designs is provided in Section 4.2.4.2 and Section 4.5.2. Due to the speed of the near wellbore streamline model a large number of runs is not an issue. Therefore, as opposed to augmenting the design for more runs when more complex curvature is observed, this experiment is set up to handle a quartic response from the beginning. This results in a total of 36 required runs. The run combinations and results are provided in Appendix IV.

4.6.3 Statistical Analysis

As with Case Studies 1 and 2, the statistical analysis is broken down into two sections. In Section 4.6.3.1 the flow rate response is analyzed, while in Section 4.6.3.2 the skin response is analyzed.

4.6.3.1 Flow Rate Response

The first part of the analysis involves determining which model best suites the response. A fit summary of each available model type is provided in Table 21. Two coefficients of determination are provided to assess the adequacy of the match.

Table 21 – Case Study 3: Flow Rate: Fit Summary

Model	Sequential P-value	Adjusted R-Squared	Predicted R-Squared	Program Recommendation
Linear	< 0.0001	0.9556	0.9486	
2FI	0.8331	0.9524	0.9370	
Quadratic	< 0.0001	0.9987	0.9978	
Cubic	< 0.0001	0.9997	0.9989	Suggested
Quartic	0.0216	1.0000		Suggested
Fifth				Aliased

The table shows R-squared values greater the 0.99 for quadratic models and above. Design Expert™ suggest using either a cubic or quartic model. However, as learned in Case Study 2, the program suggestion is not always the best solution. Often, especially for numerical simulations, the suggestion may lead to over complication within the model. As can be observed in the fit summary, the quadratic model provides very similar R-squared values, and further analysis proves it to be adequate. Therefore, a quadratic response surface is used to match the flow rate response.

An ANOVA is performed to determine which factors, interactions and corresponding higher order terms affect the response. A manual selection process is utilized, eliminating terms for F-values less than 10. The results of the ANOVA, showing only the significant factors, are provided in Table 22.

Table 22 – Case Study 3: Flow Rate: ANOVA Results (Quadratic Model)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Model	93321	2	46660	4285	< 0.0001
C – Angle of Coverage	88478	1	88478	8125	< 0.0001
C^2	3643	1	3643	335	< 0.0001
Residual	359	33	10.9		
Cor Total	93680	35			

The ANOVA reveals that only term C, the angle of coverage, is significant to the flow rate response. In addition, the second order term is also significant, providing a quadratic model of a single variable. The remaining terms concerning thickness and permeability do provide some influence upon the results, but their importance is overshadowed by the magnitude of the influence of the angle of coverage variable. Therefore, their influence is deemed insignificant in this research.

The model created has an R-squared of 0.9962, an adjusted R-squared of 0.9959 and a predicted R-squared of 0.9953. These values are very high and are in reasonable agreement, indicating a valid model.

To ensure the model is valid the underlying assumptions within the ANOVA process must be confirmed. These include normality, constant variance and randomness of the residuals. General diagnostics test are run to check these assumptions. The results of these tests are provided in Figure 71.

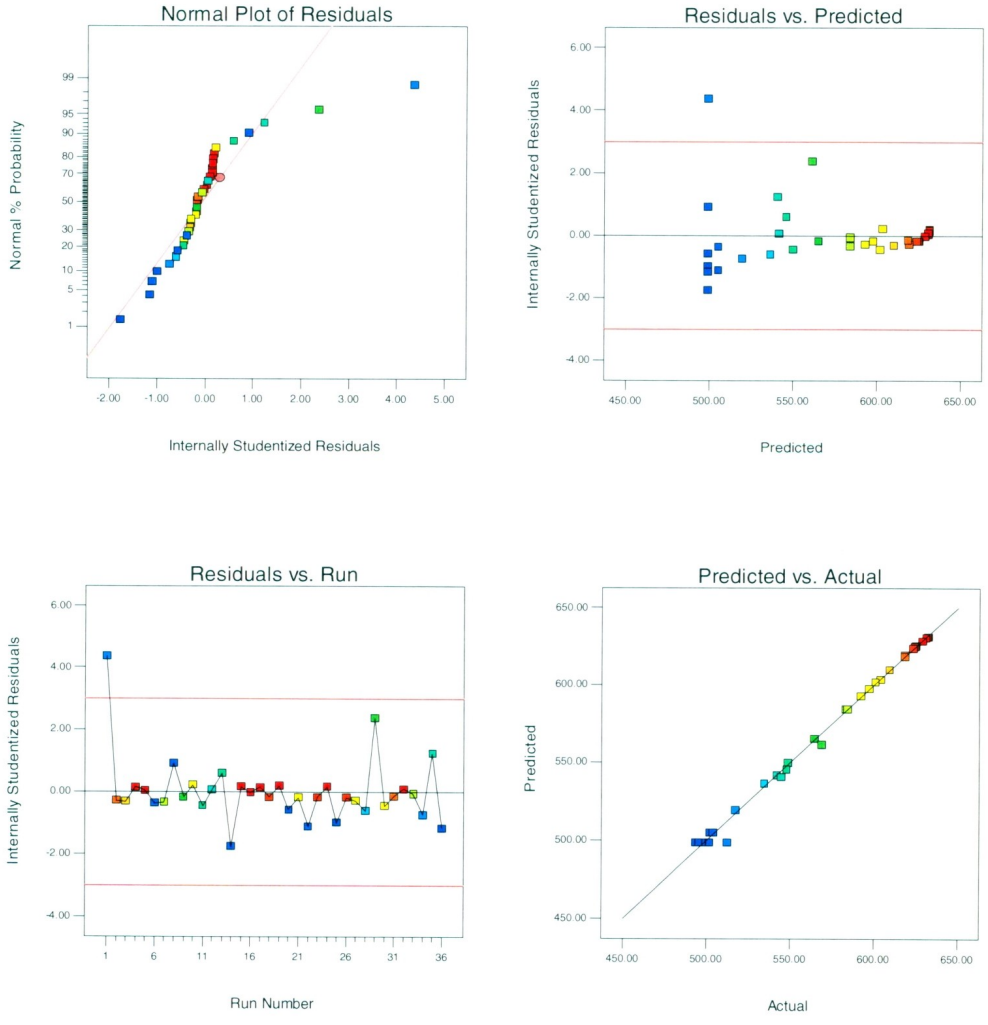


Figure 71 - Case Study 3: Flow Rate: ANOVA Assumptions Plots

The diagnostic plots reveal non-ideal results. A slight S-shaped pattern can be seen within the normal probability plot, a coning trend of decreasing variance with increasing predicted values can be seen in the residual versus predicted value plot, and an outlier exists as evidenced in several of the plots. These results are not ideal, but they do not represent an invalid model. The ANOVA analysis is best suited for physical experiments in which a measure of natural variance can be determined. In numerical simulations, no such variance exists. The error within a model can only be calculated based upon the residuals, and not the natural error within the experiment. Valid ANOVA assumptions are desirable, but are not a necessity for computer experiments. Therefore, the ANOVA process is more subjective, and may lead to results as seen here. Nevertheless, the ANOVA process remains valid, and the model can be trusted.

The meta-model as a result of this analysis is as follows:

$$Q = 632.97 + 6.1606 \times 10^{-2}\theta - 2.0681 \times 10^{-3}\theta^2$$

For this equation to be valid the angle of coverage must be in degrees, resulting in a flow rate in bbl/d. Also, recall that the model is valid for local analysis only; it is not valid outside of the range under study. As well, it will not be valid if any of the numerous constants within the model are changed. Nevertheless, it is valuable for use in understanding the behaviour of the model.

Trusting in the validity of the model, the behavior may be analyzed. Figure 72 provides a plot of the flow rate response. Since the response is a function of one variable, a regular Cartesian plot is provided.

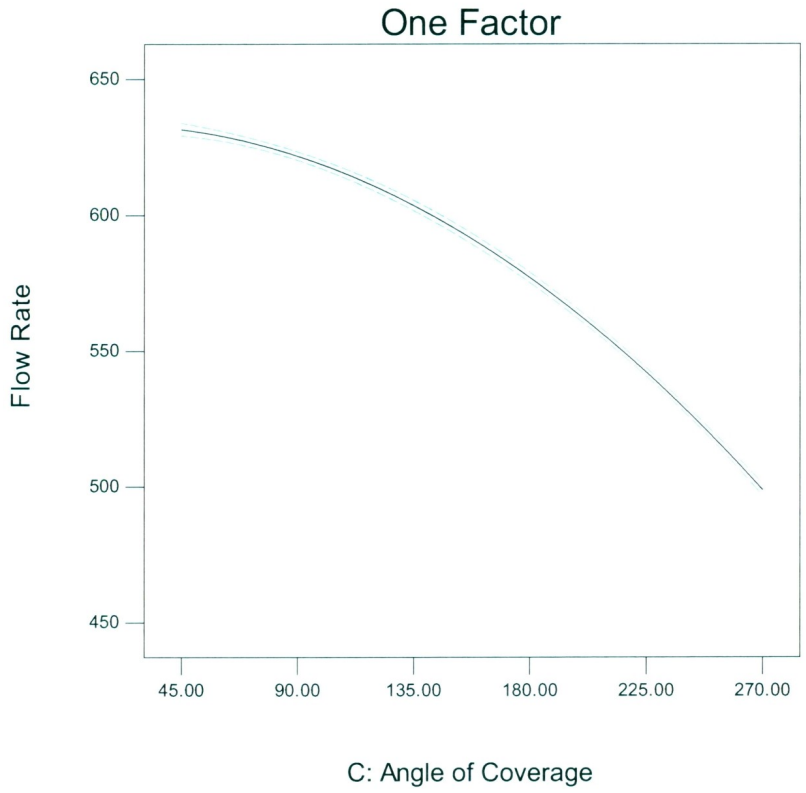


Figure 72 - Case Study 3: Flow Rate: Quadratic Model Plot

The figure clearly shows the quadratic behavior of the model, with error bounds indicating the expected variance observed from the experimental results. As the angle of coverage increases, meaning more filter cake exists on the interior walls of the open-hole completion, the productivity of the well decreases. With further increases in the angle of coverage, the magnitude of its influence on flow rate also increases, resulting in a more significant decrease in productivity.

Figure 73 provides additional insight into the model and its match to the experiment data. The significance of the outliers noticed within the ANOVA diagnostic plots can be

observed. In addition, by having the data and model shown together, a better feeling for the adequacy of the match can be obtained.

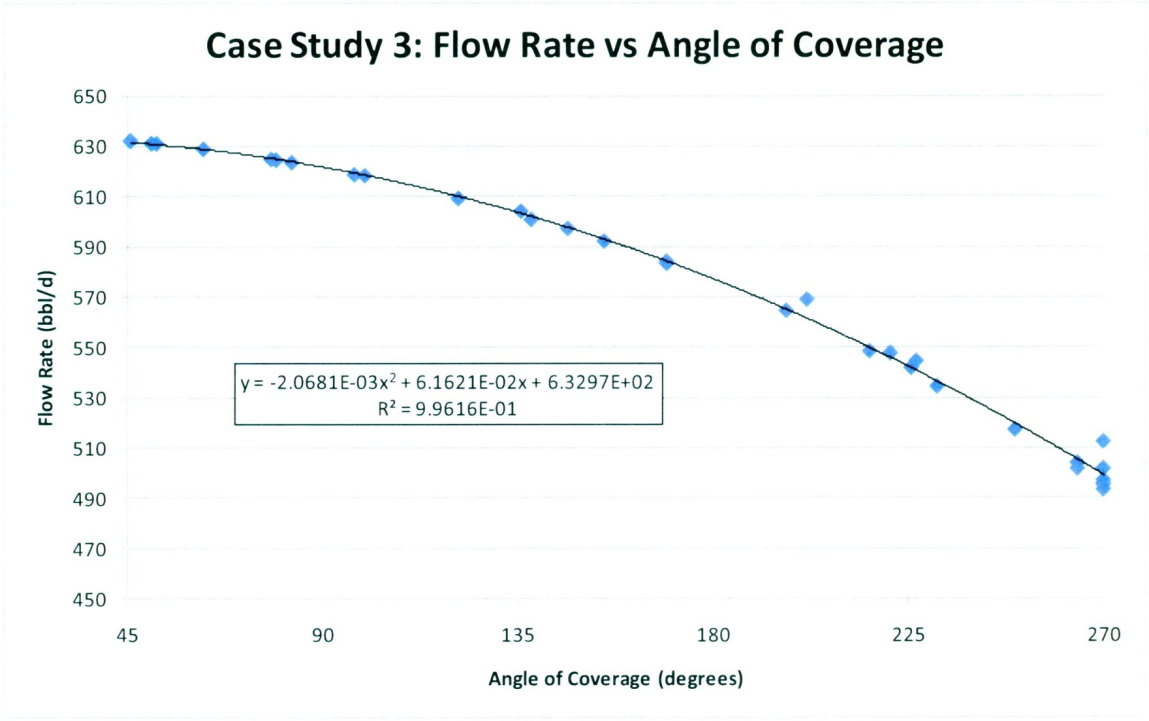


Figure 73 - Case Study 3: Flow Rate: Quadratic Model and Data

The plot reveals an excellent match between the model and experiment data. The largest outliers result in an error of less than 2%. This error is small, and well within the realm of error expected as a result of the numerous assumptions made throughout the modeling process. The angle of coverage has a quadratic relationship with flow rate. As the angle of coverage increases, the flow rate decreases. As well, the magnitude of the slope increases as the angle of coverage increases, having a more significant impact on production with greater area of coverage.

4.6.3.2 Skin Response

Table 23 provides a fit summary to help determine the most appropriate model for the skin response. The coefficients of determination as well as the sequential P-value are used to assess the adequacy of the match.

Table 23 – Case Study 3: Skin: Fit Summary

Model	Sequential P-value	Adjusted R-Squared	Predicted R-Squared	Program Recommendation
Linear	< 0.0001	0.9279	0.9166	
2FI	0.8620	0.9224	0.8977	
Quadratic	< 0.0001	0.9964	0.9944	
Cubic	< 0.0001	0.9993	0.9977	Suggested
Quartic	0.0216	1.0000		
Fifth				Aliased

The table shows R-squared values greater than 0.99 for quadratic models and above. As with the flow rate response, Design Expert™ suggests using a cubic model. However, this is likely an over complication. The quadratic model provides very similar R-squared values, was sufficient for the flow rate response, and further analysis proves it to be adequate. Therefore, a quadratic response surface is used to match the skin response.

An ANOVA is performed to determine which factors, interactions and corresponding higher order terms affect the skin response. A manual selection process is utilized, eliminating terms for F-values less than 10. The results of the ANOVA, showing only the significant factors, are provided in Table 22.

Table 24 – Case Study 3: Skin: ANOVA Results (Quadratic Model)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Model	17.0	2	8.5	2499	< 0.0001
C – Angle of Coverage	15.7	1	15.7	4606	< 0.0001
C^2	1.1	1	1.1	313	< 0.0001
Residual	0.1	33	0.0034		
Cor Total	17.1	35			

The ANOVA results coincide with the results observed in the flow rate analysis. Only term C, the angle of coverage, and its corresponding squared term are significant to the skin response. The model has an R-squared of 0.9934, an adjusted R-squared of 0.9930 and a predicted R-squared of 0.9920. These values are very high and are in reasonable agreement, indicating a valid model.

To verity the assumptions of the ANOVA several general diagnostic tests are completed. The results of these tests are provided in Figure 74.

As with the flow rate ANOVA diagnostic plots, the results are non-ideal. An S-shaped trend is observed in the normal probability plot, increasing variance with increasing predicted values is observed, and outliers exist. However, as explained in the flow rate analysis, the results of the ANOVA diagnostics are not as significant since this is a computer experiment. No natural variation exists, and therefore the error term is strictly as a result of the insignificant residual terms. Thus, despite the non-ideal trends, the model can still be trusted.

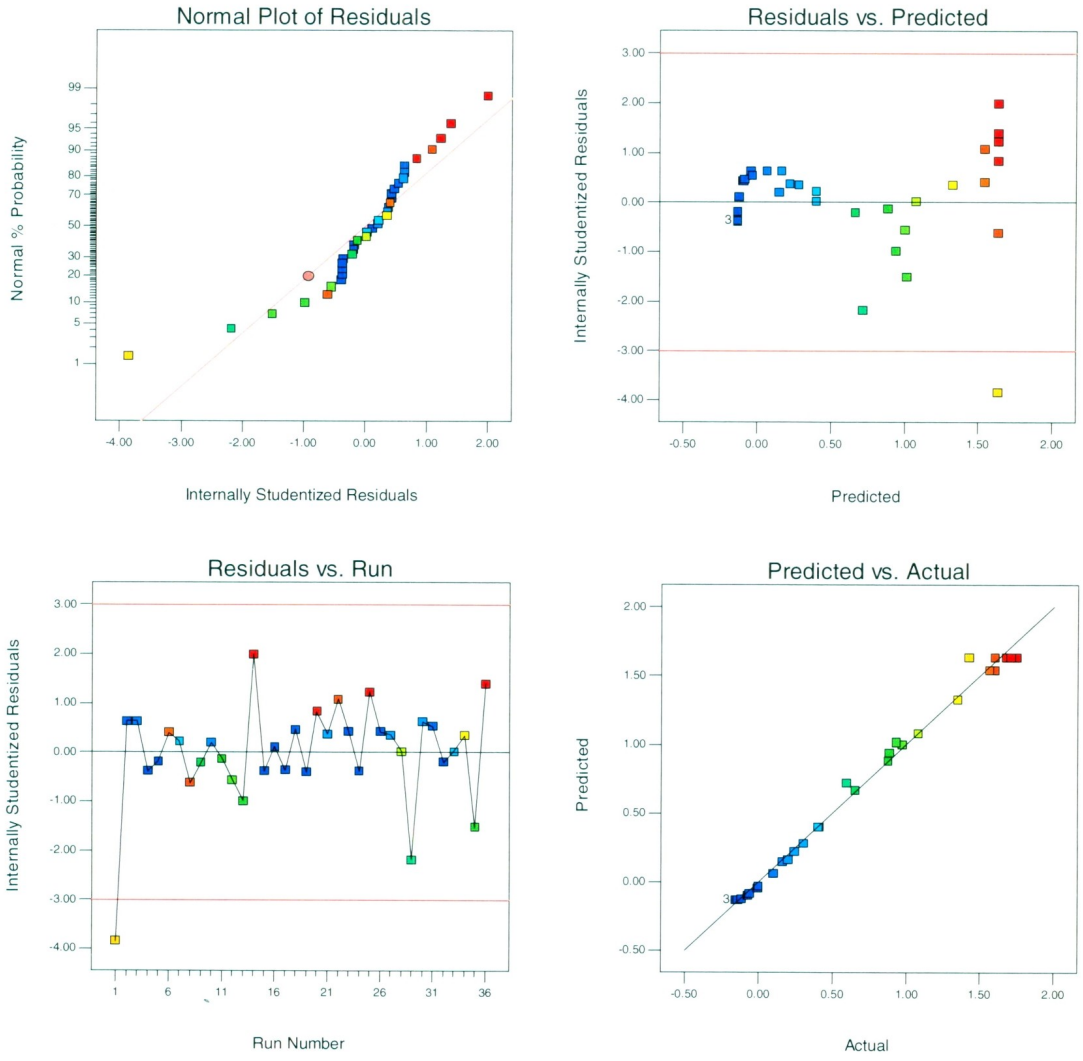


Figure 74 - Case Study 3: Skin: ANOVA Assumptions Plots

The meta-model as a result of this analysis is as follows:

$$Skin = -0.05699 - 3.2843 \times 10^{-3}\theta + 3.5362 \times 10^{-5}\theta^2$$

Again note that for this equation to be valid the angle of coverage must be in degrees, with skin being unitless. The model is valid for local analysis only, within the given range of the angle of coverage and under the same constants assumed for the case study.

Outside of these constraints the model is invalid. Nevertheless, it is valuable for use in understanding the behaviour of the model.

Trusting in the validity of the model, the behavior may be analyzed. Figure 75 provides a plot of the skin response. As with the flow rate response, the skin response is a function of one variable. Therefore a two dimensional Cartesian plot is sufficient to show the behavior of the model.

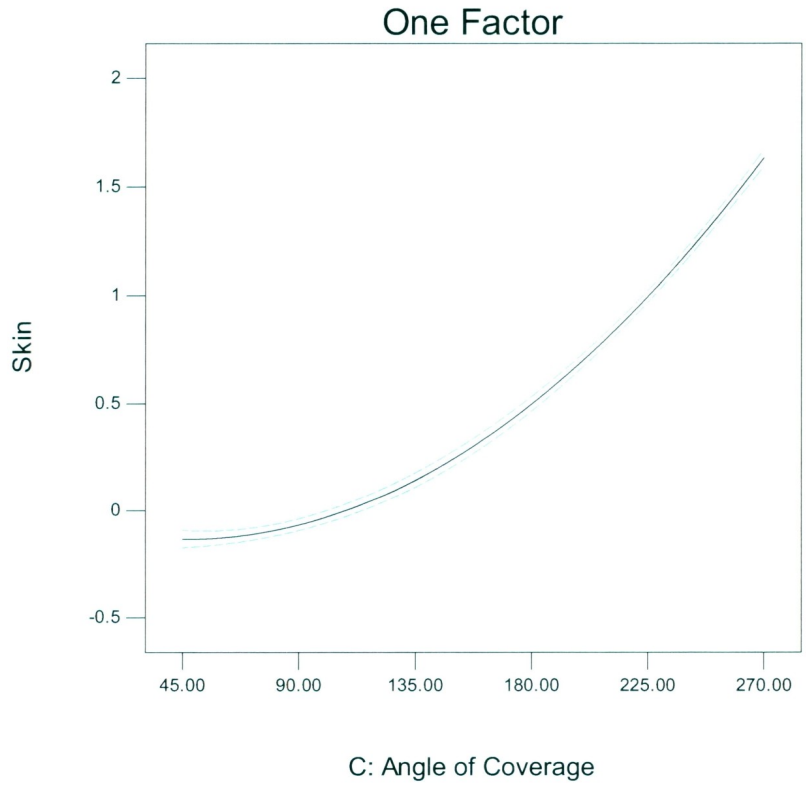


Figure 75 - Case Study 3: Skin: Quadratic Model Plot

The quadratic nature of the model is evident within the figure, with error bounds indicating the expected variance observed from the experimental results. As the angle of coverage increases, the total skin increases, thereby decreasing the productivity. With

further increases in the angle of coverage, the magnitude of its influence on skin also increases, resulting in more significant decreases in productivity. This result concurs with the result of the flow rate analysis.

Figure 76 is provided to assess the model in relation the experiment results.

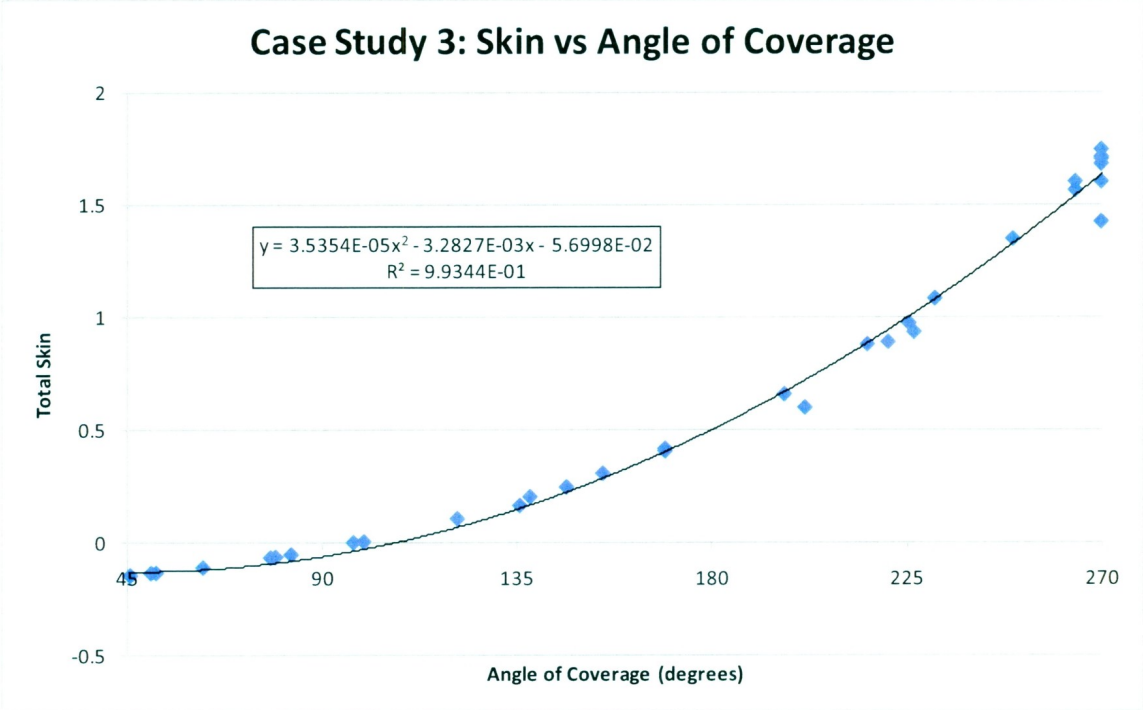


Figure 76 - Case Study 3: Skin: Quadratic Model and Data

As can be seen in the figure, an excellent match is obtained, confirming the conclusion regarding the ANOVA results. The outliers to the far right result in a 12% error, within acceptable limits. The model confirms the significance of the angle of converge in influencing the apparent skin. The other terms under study relating to the properties of the filter cake have a minimal effect in comparison, and are concluded to be insignificant over the range investigated.

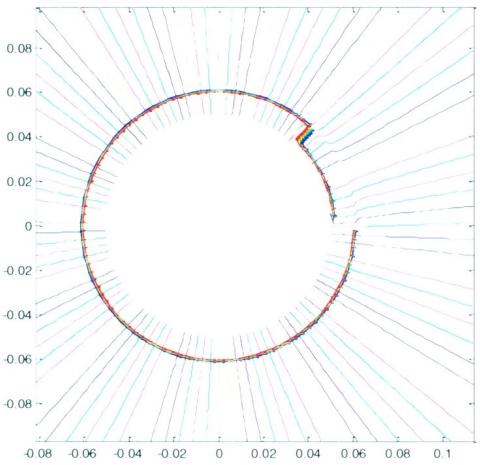
4.6.4 Conclusions

In this case study an IV-optimal design was used to determine the nature of various filter cake properties in an open-hole completion within a near wellbore streamline model. Both flow rate and skin responses were used to assess the impact on the productivity of a well. For both responses only the angle of coverage proved to be significant. Two quadratic models were created, both of which provided excellent matches to the experiment data.

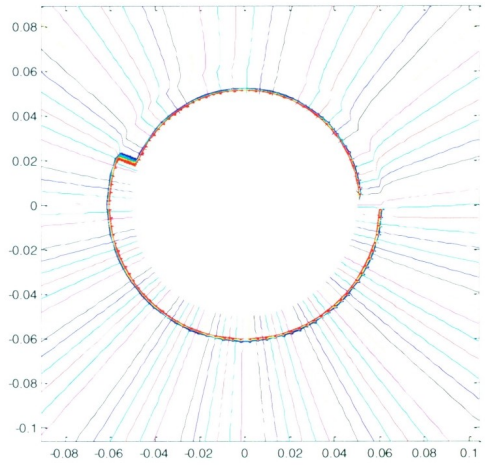
The other factors under study, those being filter cake thickness and filter cake permeability, both proved to be insignificant over the range studied. Their influence results in the variation between data points and the model, but their impact is small in comparison to the importance of the angle of coverage. Therefore, over the range of values under study, it can be concluded that the properties of the filter cake are insignificant. It is the mere presence and lateral coverage of filter cake on the internal face of the wellbore that affects the productivity of a well, regardless of the properties of the filter cake itself. The extremely low permeability of filter cake means that its mere presence causes a change in flow. Therefore, changes in permeability and thickness have no effect.

Throughout this analysis the effect of various parameters has been determined, but little focus has been given into the nature of the streamline distributions. Figure 77 shows the effect of the angle of coverage of filter cake on the streamline distribution within the near wellbore model.

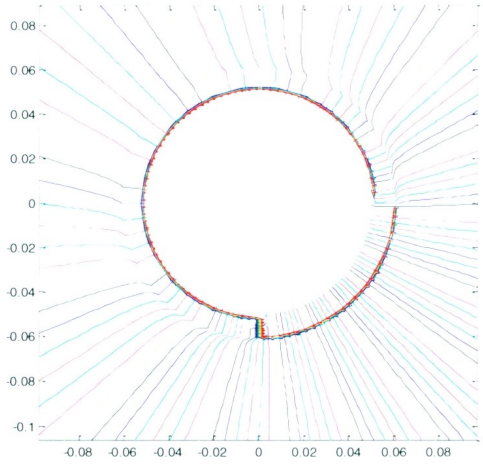
Three subplots are provided in the figure, one for 45°, 157.5° and 270° of coverage, respectively. The coverage can be difficult to observe. It always begins at zero degrees and moves counterclockwise around the wellbore to its end location. A circular line distinguishing the transition from the bulk reservoir face to "free space" can be observed. Where this line does not exist, or where it cuts into the internal boundary, represents the presence of filter cake.



45°



157.5°



270°

Figure 77 - Influence of Angle of Coverage on Streamline Distribution

The effect of the angle of coverage is evident within the figure. Where no filter cake exists, streamline move radially towards the wellbore. Streamlines which approach the filter cake tend to deflect in the direction of the nearest region of high permeability. Some streamlines may reach the high permeability region. However, many are forced to hit and travel through the filter cake, resulting in reduced productivity. Due to the Laplace procedure, a pressure differential will exist between the reservoir nodes and adjacent radial filter cake nodes. As a result, a velocity will exist at the face. Streamline approach the filter cake and begin to deflect away. However, if the streamline touches the filter cake (edge of the grid block) it is forced to travel through it as a result of the Pollock streamline tracking methodology.

As the angle of filter cake increases, more streamlines are forced to enter the filter cake, resulting in decreased flow and higher skin. The deflection of the streamlines can be observed by analyzing the density of the streamlines at the internal boundary. As most evident in the 270° subfigure, streamlines tend to enter with high density in the area in which no filter cake exist. The streamlines are more spaced, having less density, in the areas in which filter cake is present. This is as a result of streamlines deflecting further out in the reservoir domain to avoid the low permeability section. Figure 78 provides a plot showing the curvature further out in the reservoir domain. The curvature of the streamlines and the progression through the filter cake is responsible for the reduction in the productivity.

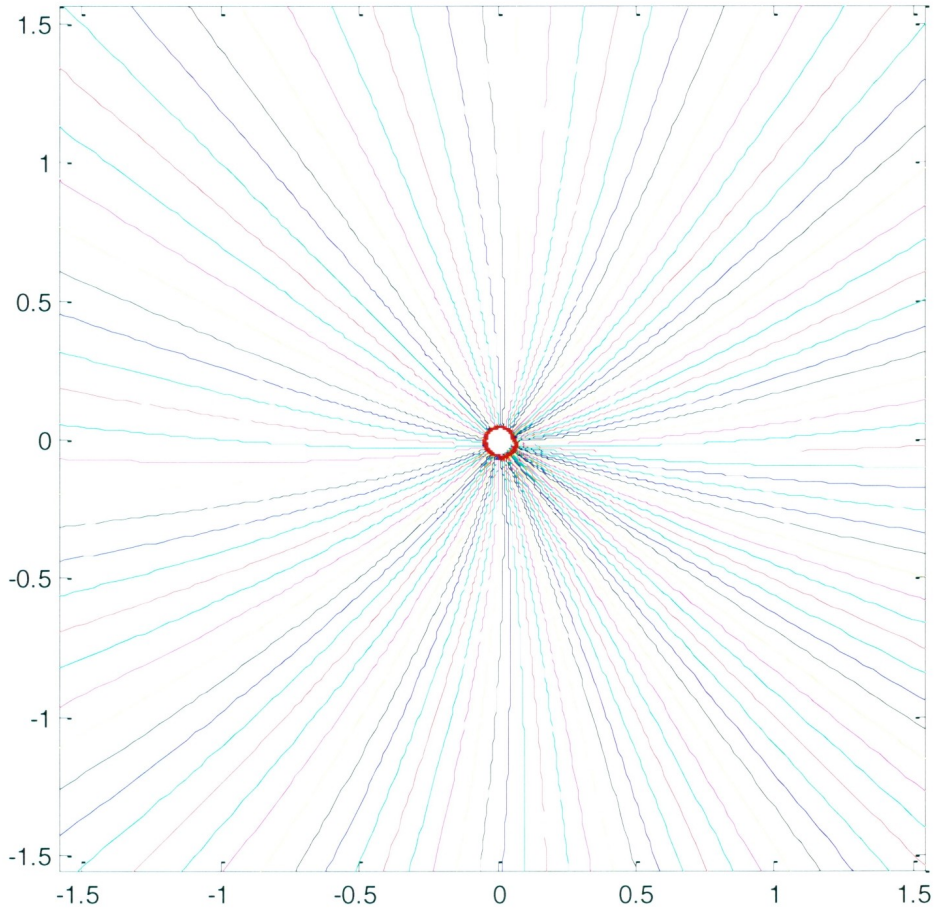


Figure 78 - Curvature as a Result of Angle of Coverage (270°)

4.7 Investigating Non-Darcy Flow Effects

The modeling process presented in this research deals strictly with Darcy flow. For this to be valid, flow throughout the reservoir medium must remain laminar. While this is clearly the case for slow moving areas in the outer regions of the reservoir domain, the increase in velocity with decreasing radius leads to the possibility for non-laminar flow in the very near wellbore vicinity. If the flow regime becomes non-laminar, non-Darcy flow effects must be considered. In the industry, non-Darcy flow effects occur predominantly in high productivity gas wells.

To determine if the flow is laminar for the cases studied, the Reynolds number can be calculated. The Reynolds number is a dimensionless number that compares the ratio of inertial forces to viscous forces. It is commonly used in fluid mechanics to distinguish between various flow regimes; laminar, transition and turbulent. The equation for the Reynolds number is as follows:

$$R_e = \frac{\rho V D}{\mu}$$

where ρ is the fluid density, V is the fluid velocity, D is the hydraulic diameter and μ is the fluid viscosity. In traditional pipe flow methodologies laminar flow is considered to have a Reynolds number of less than 2300. The additional complexity associated with the flow of fluids through porous media reduces this threshold to less than 10.

As opposed to determining the Reynolds number for each case, typical values used within each cased study are used to determine the maximum flow rate for laminar flow.

Providing the flow rates occur under this value, the case study results should be valid. The values used to determine the threshold are provided in Table 25.

Table 25 – Darcy Flow Threshold Values

Parameter	Value	Units
Wellbore Radius	0.05	m
Reservoir Layer Thickness	5	m
Porosity	0.25	
Oil Density	800	kg/m^3
Oil Viscosity	$8x10^{-4}$	Pa s
Pore Throat Diameter	$2x10^{-6}$	m

Note that the chosen pore throat diameter comes from an article by Philip Nelson in the American Association Petroleum Geologists (AAPG) Bulletin (Nelson, 2009). The remainder of the values are selected based upon typical values used in the previous case studies.

For the calculation, the flow rate represents the flow into the well. Area represents the cylindrical reservoir region connected to the well. The formula for the area of the cylindrical reservoir wall is as follows:

$$A = 2\pi r_w h$$

Note that Darcy velocity is:

$$u = \frac{Q}{A}$$

To determine the real velocity the Darcy velocity must be divided by the formation porosity:

$$V = \frac{u}{\emptyset} = \frac{Q}{A\emptyset} = \frac{Q}{2\pi r_w h\emptyset}$$

The Reynolds number expression may be reduced as follows:

$$R_e = \frac{\rho V D}{\mu} = \frac{\rho\left(\frac{Q}{2\pi r_w h \emptyset}\right) D}{\mu} = \frac{\rho Q D}{2\pi r_w h \emptyset \mu}$$

Rearranging the expression:

$$Q = \frac{2\pi r_w h \emptyset \mu R_e}{\rho D}$$

Inserting the chosen values:

$$Q = \frac{2\pi(0.05m)(5m)(0.25)(8 \times 10^{-4} Pa\ s)(10)}{(800 kg/m^3)(2 \times 10^{-6} m)} = 0.01964 \frac{m^3}{s} = 10670 \frac{bbl}{d}$$

Therefore using the conservative approach of laminar flow ending at a Reynolds number of 10 the maximum flow rate is $10670\, bbl/d$. All case study results presented in this research are significantly less than this maximum value, eliminating the need for concern in relation to non-Darcy flow effects. Additional discussion regarding non-Darcy flow effects is provided in the recommendations for future work in Chapter 6.

- Chapter 5 -

5.0 Extension to Three Dimensional Streamline Modeling

5.1 Introduction

In Chapter 3 the procedure for streamline modeling in two dimensions is explained in detail, outlining the major steps involved within the modeling process. This includes details on working in polar coordinates, grid discretization, solving the Laplace equation, utilizing Pollock's methodology to generate streamlines based on time of flight and using stream tubes to determine flow rate. The model developed in Chapter 3 focuses on a heterogeneous and isotropic medium in which reservoir heterogeneity is the dominating factor influencing flow profiles. The novelty of the approach includes the first fully functional use of streamlines in a radial geometry as well as proving streamline modeling to be stable within the near wellbore region. In Chapter 4 the model is applied to realistic case studies, proving its usefulness and validity.

The extension of streamline modeling to three dimensions has been established in the literature for over 15 years, as discussed in Chapter 2. The focus of this chapter is to explain the process for modeling streamlines in a three dimensional cylindrical coordinate system. A three dimensional representation of a cylindrical grid is provided in Figure 79.

The three dimensional modeling process explained in this chapter is provided from a higher level perspective as compared to that of Chapter 3 since most of the derivations and modeling methodology will not differ from that previously explained. The

appropriate modifications to the modeling procedures are discussed, including Laplace equation alterations, streamline modeling alterations and a new method to determine flow rate. The focus is on a heterogeneous isotropic medium, with discussion on the general case of a heterogeneous anisotropic medium. Detailed programming in three dimensions is not performed, but the procedure is established with several recommendations for future research being provided in Chapter 6.

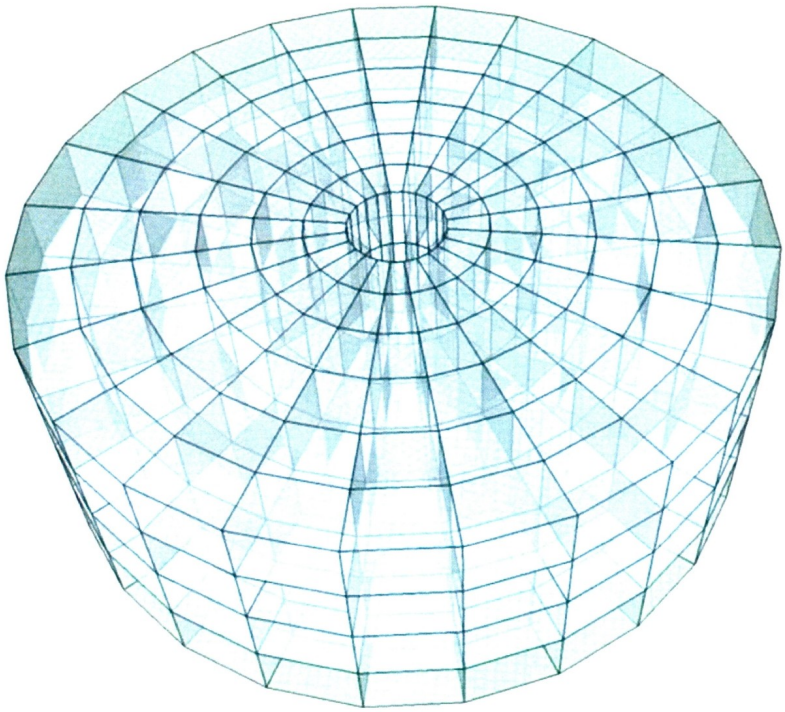


Figure 79 - 3D Cylindrical Grid Layout

For consistency, the model elaborated upon in this chapter remains a vertical well in a cylindrical reservoir. The advantage of modeling in three dimensions for this scenario lies in the ability to model vertical flow movement within the reservoir as opposed to assuming strictly parallel flow for the horizontal planes. Therefore, the assumption of a

fully perforated layer for the entire thickness of the reservoir slice may be lifted, allowing flow profiles to veer from parallel with the horizontal plane. In removing this assumption accuracy is increased and more realistic completions of increased complexity may be investigated.

The most significant benefit of three dimensional modeling comes from its applicability to all reservoir and well configurations. The modeling process may be applied to any reservoir domain with various boundary conditions and fault interactions under either polar or Cartesian coordinate systems. Additionally, horizontal and deviated wells may be modeled. The versatility of the modeling process allows for full field streamline modeling, allowing greater insight for reservoir engineers in understanding the intricacies of flow in complex reservoirs.

The remainder of this chapter explains the modeling alterations required to extend the two dimensional radial scenario for a vertical well to a three dimensional cylindrical reservoir.

5.2 Laplace Equation Alterations

To extend the work of Chapter 3 to three dimensions, the vertical direction must be entered into the formulation process. Therefore, the polar system must be altered to a cylindrical system, incorporating the z-axis as the third dimension in addition to radius and angle.

Recall from Section 3.3 the unit vector notation for a polar coordinate system is:

$$\vec{e}_r = (\cos\theta\,, \sin\theta)$$

$$\vec{e}_\theta = (-\sin\theta\,, \cos\theta)$$

The incorporation of the z-direction is a straight forward process since the z-axis corresponds with a principle permeability direction (Johansen, 2008). The unit vector notation in three dimensions becomes:

$$\vec{e}_r = (\cos\theta\,, \sin\theta\,, 0)$$

$$\vec{e}_\theta = (-\sin\theta\,, \cos\theta\,, 0)$$

$$\vec{e}_z = (0,0,1)$$

Recall that the polar notation for Darcy's law, as derived in Section 3.3.2, is:

$$\vec{u} = -\frac{1}{\mu}\bar{K}\nabla_{\mathrm{r},\theta}p = -\frac{1}{\mu}\begin{bmatrix}K_r & K_\theta \\ K_\theta & K_t\end{bmatrix}\begin{bmatrix}\dfrac{\partial p}{\partial r} \\ \dfrac{1}{r}\dfrac{\partial p}{\partial \theta}\end{bmatrix}$$

Again, the extension of this to three dimensions is a simple application as a result of the alignment of the z-axis. In cylindrical coordinates Darcy's law becomes:

$$\vec{u} = -\frac{1}{\mu}\bar{K}\nabla_{r,\theta}p = -\frac{1}{\mu}\begin{bmatrix} K_r & K_\theta & 0 \\ K_\theta & K_t & 0 \\ 0 & 0 & K_z \end{bmatrix}\begin{bmatrix} \frac{\partial p}{\partial r} \\ \frac{1}{r}\frac{\partial p}{\partial \theta} \\ \frac{\partial p}{\partial z} \end{bmatrix}$$

The derivation of the Laplace equation in polar coordinates is provided in Section 3.3.3. To apply this to three dimensions the pressure derivative in the z-direction must be considered. The three dimensional derivation is as follows:

$$\nabla_{r,\theta,z} \cdot \vec{u} = \frac{1}{r}\frac{\partial}{\partial r}(r\, u_r) + \frac{1}{r}\frac{\partial}{\partial \theta}(u_\theta) + \frac{\partial}{\partial z}(u_z) = 0$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\left(K_r\frac{\partial p}{\partial r} + \frac{1}{r}K_\theta\frac{\partial p}{\partial \theta}\right)\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(K_\theta\frac{\partial p}{\partial r} + \frac{1}{r}K_t\frac{\partial p}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(K_z\frac{\partial p}{\partial z}\right) = 0$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(rK_r\frac{\partial p}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(K_\theta\frac{\partial p}{\partial \theta}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(K_\theta\frac{\partial p}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(K_t\frac{\partial p}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(K_z\frac{\partial p}{\partial z}\right) = 0$$

This is the generalized form of the Laplacian in a three dimensional cylindrical coordinate system for the case where the Cartesian coordinate system aligns with the principle permeability directions. As with the two dimensional polar approach, simplifications can be made for homogenous or isotropic media. For a heterogeneous isotropic case this reduces to:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(rK_r\frac{\partial p}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(K_t\frac{\partial p}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(K_z\frac{\partial p}{\partial z}\right) = 0$$

For a homogenous anisotropic case the Laplacian reduces to:

$$K_r\frac{\partial^2 p}{\partial r^2} + \frac{K_r}{r}\frac{\partial p}{\partial r} + 2\frac{K_\theta}{r}\frac{\partial^2 p}{\partial r \partial \theta} + \frac{K_t}{r^2}\frac{\partial^2 p}{\partial \theta^2} + K_z\frac{\partial^2 p}{\partial z^2} = 0$$

For the simplest case of a homogenous and isotropic medium the Laplacian reduces to:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} + \frac{1}{r^2}\frac{\partial^2 p}{\partial \theta^2} + \frac{\partial^2 p}{\partial z^2} = 0$$

The solution of the Laplace equation, regardless of the case chosen, follows the same general approach outlined in Section 3.4. Following discretization, a set of linear equations must be generated followed by the application of appropriate boundary conditions. Following the correct setup in matrix format the solution may be generated.

5.3 Streamline Modeling Alterations

Once the solution to the Laplace equation is obtained, the pressure at all nodes throughout the reservoir medium is known. Knowing the nodal pressures, the velocities at all face boundaries can be determined. For a sector in three dimensions six face velocities are required; two for the radial faces, two for the tangential faces, and two for the plane faces. The required face velocities, along with proper three dimensional notations, are provided in Figure 80.

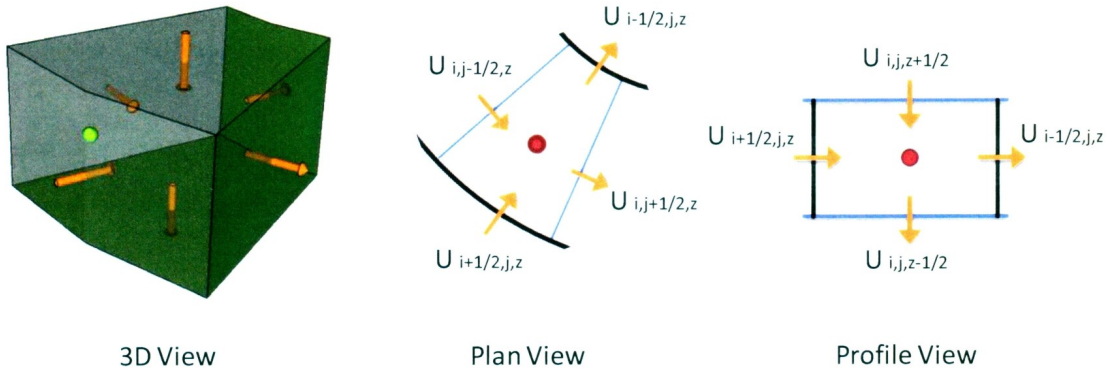


Figure 80 - Sector Velocity Field in 3D

The equations for the six velocity terms are as follows:

$$u_r = -\frac{1}{\mu}\left(K_r \frac{\partial p}{\partial r} + \frac{1}{r} K_\theta \frac{\partial p}{\partial \theta}\right)$$

$$u_\theta = -\frac{1}{\mu}\left(K_\theta \frac{\partial p}{\partial r} + \frac{1}{r} K_t \frac{\partial p}{\partial \theta}\right)$$

$$u_z = -\frac{1}{\mu}\left(K_z \frac{\partial p}{\partial z}\right)$$

Therefore, in discretized form for the general case of a heterogeneous and anisotropic medium the flux formulations become:

$$u^r_{i+\frac{1}{2},j,k} = \lambda^r_{i+\frac{1}{2},j,k}\frac{p_{i+1,j,k}-p_{i,j,k}}{r_{i+1}-r_i} + \frac{1}{2r_{i+1/2}}\left(\lambda^\theta_{i,j+\frac{1}{2},k}\frac{p_{i,j+1,k}-p_{i,j,k}}{\Delta\theta} + \lambda^\theta_{i,j-\frac{1}{2},k}\frac{p_{i,j,k}-p_{i,j-1,k}}{\Delta\theta}\right)$$

$$u^r_{i-\frac{1}{2},j,k} = \lambda^r_{i-\frac{1}{2},j,k}\frac{p_{i,j,k}-p_{i-1,j,k}}{r_i-r_{i-1}} + \frac{1}{2\,r_{i-1/2}}\left(\lambda^\theta_{i,j+\frac{1}{2},k}\frac{p_{i,j+1,k}-p_{i,j,k}}{\Delta\theta} + \lambda^\theta_{i,j-\frac{1}{2},k}\frac{p_{i,j,k}-p_{i,j-1,k}}{\Delta\theta}\right)$$

$$u^\theta_{i,j+\frac{1}{2},k} = \frac{1}{r_i}\lambda^t_{i,j+\frac{1}{2},k}\frac{p_{i,j+1,k}-p_{i,j,k}}{\Delta\theta} + \frac{1}{2}\left(\lambda^\theta_{i,j+\frac{1}{2},k}\frac{p_{i,j+1,k}-p_{i,j,k}}{r_{i+1}-r_i} + \lambda^\theta_{i,j-\frac{1}{2},k}\frac{p_{i,j,k}-p_{i,j-1,k}}{r_i-r_{i-1}}\right)$$

$$u^\theta_{i,j-\frac{1}{2},k} = \frac{1}{r_i}\lambda^t_{i,j-\frac{1}{2},k}\frac{p_{i,j,k}-p_{i,j-1,k}}{\Delta\theta} + \frac{1}{2}\left(\lambda^\theta_{i,j+\frac{1}{2},k}\frac{p_{i,j+1,k}-p_{i,j,k}}{r_{i+1}-r_i} + \lambda^\theta_{i,j-\frac{1}{2},k}\frac{p_{i,j,k}-p_{i,j-1,k}}{r_i-r_{i-1}}\right)$$

$$u^z_{i,j,k+\frac{1}{2}} = \lambda^z_{i,j,k+\frac{1}{2}}\frac{p_{i,j,k+1}-p_{i,j,k}}{z_{i,j,k+1}-z_{i,j,k}}$$

$$u^z_{i,j,k-\frac{1}{2}} = \lambda^z_{i,j,k-\frac{1}{2}}\frac{p_{i,j,k}-p_{i,j,k-1}}{z_{i,j,k}-z_{i,j,k-1}}$$

Note that i, j and k are the counters in the r, θ and z directions respectively. Also note that $z_{i,j,k+1} - z_{i,j,k}$ or $z_{i,j,k} - z_{i,j,k-1}$ may be replaced with Δz provided a uniform layer thickness is chosen.

The specialized case of a heterogeneous isotropic medium, as modeled in Chapter 3, may be reduced as a result of $K_\theta = 0$. Therefore the flux formulations become:

$$u^r_{i+\frac{1}{2},j,k} = \lambda^r_{i+\frac{1}{2},j,k} \frac{p_{i+1,j,k} - p_{i,j,k}}{r_{i+1} - r_i}$$

$$u^r_{i-\frac{1}{2},j,k} = \lambda^r_{i-\frac{1}{2},j,k} \frac{p_{i,j,k} - p_{i-1,j,k}}{r_i - r_{i-1}}$$

$$u^\theta_{i,j+\frac{1}{2},k} = \frac{1}{r_i} \lambda^t_{i,j+\frac{1}{2},k} \frac{p_{i,j+1,k} - p_{i,j,k}}{\Delta\theta}$$

$$u^\theta_{i,j-\frac{1}{2},k} = \frac{1}{r_i} \lambda^t_{i,j-\frac{1}{2},k} \frac{p_{i,j,k} - p_{i,j-1,k}}{\Delta\theta}$$

$$u^z_{i,j,k+\frac{1}{2}} = \lambda^z_{i,j,k+\frac{1}{2}} \frac{p_{i,j,k+1} - p_{i,j,k}}{\Delta z}$$

$$u^z_{i,j,k-\frac{1}{2}} = \lambda^z_{i,j,k-\frac{1}{2}} \frac{p_{i,j,k} - p_{i,j,k-1}}{\Delta z}$$

Once the face velocities are calculated the streamline modeling process can begin. Note that the velocities used above are Darcy velocities or volumetric fluxes as opposed to real velocity. These may be converted initially at this stage or after the streamlines are mapped, but it must be converted at one of these times to ensure proper time of flight values for further calculations involving flow rate, mass transport, saturations or the like.

Knowing the face velocities, the entire velocity field throughout the reservoir medium (within the grid blocks) must be generated. To calculate these values different forms of interpolation are utilized. As with the two dimensional case, velocity in the radial

direction will increase inversely proportional to the radius, which is established as follows:

$$u_r(r) = \frac{a_r}{r} + b_r$$

In the transverse and vertical directions velocity is considered linear with the following interpolation equations being applied:

$$u_\theta(\theta) = a_\theta \theta + b_\theta$$

$$u_z(z) = a_z z + b_z$$

The known face velocities are used as boundary conditions to determine the unknown coefficients $a_r, b_r, a_\theta, b_\theta, a_z$ and b_z, which will differ for each grid block. For the case of a heterogeneous isotropic medium, the equations, derived from a substitution and elimination approach, are as follows:

$$a^r_{i,j,k} = \frac{r_{i-1/2} r_{i+1/2}}{r_{i+1/2} - r_{i-1/2}} \left(u^r_{i-\frac{1}{2},j,k} - u^r_{i+\frac{1}{2},j,k} \right)$$

$$b^r_{i,j,k} = u^r_{i+\frac{1}{2},j,k} - \frac{a^r_{i,j.k}}{r_{i+1/2}}$$

$$a^\theta_{i,j,k} = \frac{u^\theta_{i,j-\frac{1}{2},k} - u^\theta_{i,j+\frac{1}{2},k}}{\theta_{j-1/2} - \theta_{j+1/2}}$$

$$b^\theta_{i,j,k} = u^\theta_{i,j+\frac{1}{2},k} - a^\theta_{i,j,k} \theta_{j+1/2}$$

$$a^z_{i,j,k} = \frac{u^z_{i,j,k-\frac{1}{2}} - u^z_{i,j,k+\frac{1}{2}}}{z_{k-1/2} - z_{k+1/2}}$$

$$b^z_{i,j,k} = u^z_{i,j,k+\frac{1}{2}} - a^z_{i,j,k} z_{k+1/2}$$

Once the velocities coefficients are determined the velocity field throughout a grid block, and the entire reservoir medium, is known. Next, the travel time of a particle must be determined. In a two dimensional radial grid only two travel times are necessary; radial and tangential. In three dimensions the vertical travel time must also be considered. The travel times in question are illustrated in Figure 81.

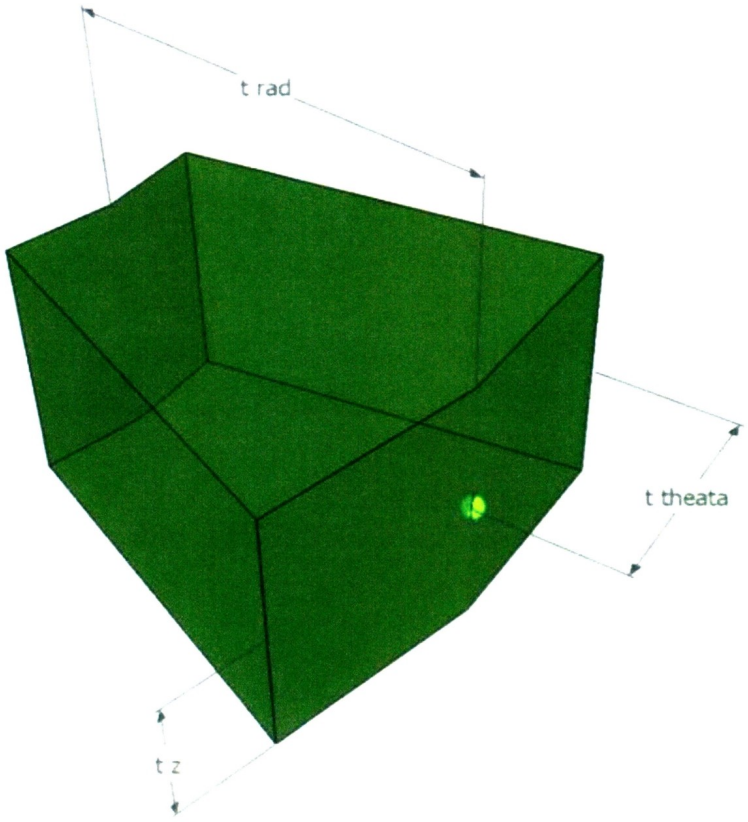


Figure 81 - Radial Sector Travel Times for 3D Modeling

The streamline process, as discussed in Chapter 3, involves calculating the travel times from an entrance point to each of the prospective exit faces. The direction with the least travel time is the exit face for the particular grid block. For three dimensions in a cylindrical system, this would be represented by:

$$t_{min} = \min(t_r, t_\theta, t_z)$$

Each of the travel times are calculated by integrating the inverse of velocity with respect to distance and evaluating from the entrance point to the prospective exit face. This is represented mathematically as follows:

$$dx = u_x(x)dt$$

$$dt = \frac{1}{u_x(x)}dx$$

$$t_x = \int_{x_{ex}}^{x_{en}} \frac{1}{u_x(x)}dx$$

In the above expression x represents a particular direction; either radial, tangential or vertical. The resultant equation for time traveled to each face is dependent upon the velocity expression.

Continuing with the case of a heterogeneous isotopic medium, the time equations for each direction are:

$$t_r = \frac{1}{{b_r}^2}\left(b_r(r_{en} - r_{ex}) - a_r ln\left(\frac{a_r + b_r r_{en}}{a_r + b_r r_{ex}}\right)\right)$$

$$t_\theta = \frac{1}{a_\theta}\left(ln(b_\theta + a_\theta\theta_{ex}) - ln(b_\theta + a_\theta\theta_{en})\right)$$

$$t_z = \frac{1}{a_z}\left(ln(b_z + a_z z_{ex}) - ln(b_z + a_z z_{en})\right)$$

As mentioned, the actual time for the particle to travel from the entrance face to the exit face, known as the time of flight, is the least of t_r, t_θ or t_z.

Once the minimum time and exit face is determined the precise location of the exit point can be established. This is performed by substituting the minimum time into the time equations for the remaining directions and solving for the respective exit coordinates. An illustration of t_θ being the minimum time is provided in Figure 82.

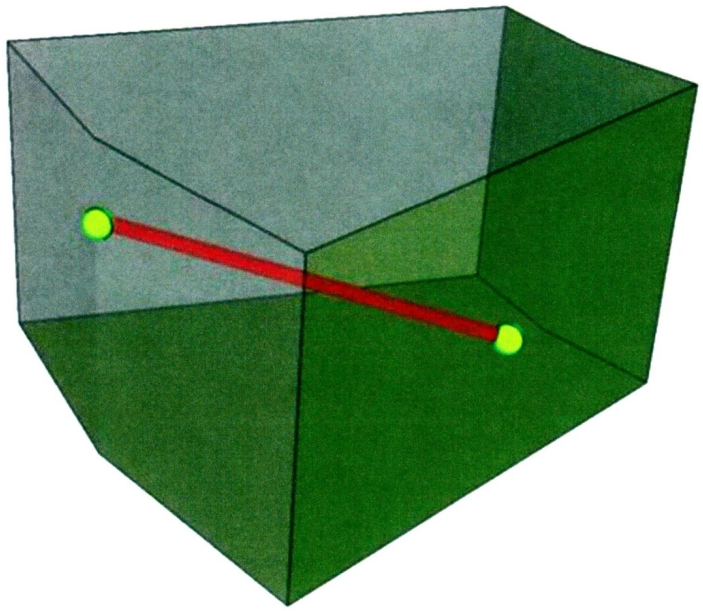


Figure 82 - Path of Particle in 3D Sector

The equations for determining the exit location for the radial and tangential directions were derived in Chapter 3 and are not modified for the three dimensional approach. As well, since the z direction undergoes the same linear interpolation as the theta direction, the derivation is similar. The equations to find the exit locations when the minimum travel time is not in the respective directions are:

$$r_{ex:\,n+1} = r_{ex:\,n} - \frac{t_{min} - \frac{1}{b_r}(r_{en} - r_{ex}) + \frac{a_r}{{b_r}^2}\left(ln(a_r + b_r r_{en}) - ln(a_r + b_r r_{ex})\right)}{\frac{1}{b_r} - \frac{a_r}{b_r}\frac{1}{a_r + b_r r_{ex}}}$$

$$\theta_{ex} = \frac{e^{t_{min} a_\theta}(b_\theta + a_\theta \theta_{en}) - b_\theta}{a_\theta}$$

$$z_{ex} = \frac{e^{t_{min} a_z}(b_z + a_z z_{en}) - b_z}{a_z}$$

Note the radial direction requires a numerical approach, while the tangential and vertical directions are closed form analytical expressions.

Once the location of the exit point is determined, it then becomes the entrance point for the adjacent block. The entire process is repeated from block to block until a streamline is generated starting from the entrance boundary (external radius) to the exit boundary (internal radius). This is then repeated for numerous external entrance points to generate a streamline distribution throughout the reservoir medium.

5.4 Flow Rate Calculation Alterations

In Chapter 3 flow rate is determined in two dimensions by tying adjacent streamlines together to form stream tubes. Area within the stream tubes is determined by calculating the magnitude of the cross products between vectors at radial face intersections and averaging for opposite angles in the quadrilateral. Knowing the area for each segment, averages for porosity and time of flight are obtained between the adjacent streamlines and flow rate may be determined via the following summation along all streamline segments:

$$q = \frac{h}{T}\sum_{i,j} \phi_{i,j} A_{i,j}$$

In general, the extension of the stream tube approach to three dimensions becomes more complex as a result of the difficulties involved in tracking tube geometries in three dimensions. Depending upon the reservoir and well geometry, the area or volume calculations for stream tubes may become difficult to solve in a systematic mathematical fashion.

However, for a cylindrical reservoir a systematic approach may be obtained utilizing radial boundary intersections and the basic velocity field principle enforcing streamlines to never cross. Section 5.4.1 discusses the procedure to determine flow rate in three dimensions for the case of a three dimensional cylindrical reservoir with a vertical well at its center. Section 5.4.2 discusses an alternative approach utilizing time of flight to establish flow rate independent of the stream tube approach.

5.4.1 Flow Rate Calculations with Three Dimensional Stream Tubes

As discussed previously, streamlines provide an instantaneous viewpoint of the velocity field throughout a reservoir. Since the velocity at a particular point can only be in one direction, streamlines can never cross. This principle is essential in the calculation procedure to follow.

For a cylindrical reservoir, streamlines must start along the external boundary and work themselves in towards the wellbore (internal boundary). The external boundary in this case is the exterior of a cylinder with a radius equal to r_e. To aid in the visualization of the stream tubes, imagine the unwrapping of the exterior boundary, or all radial rings throughout the reservoir medium, into a rectangle. This is illustrated in Figure 83.

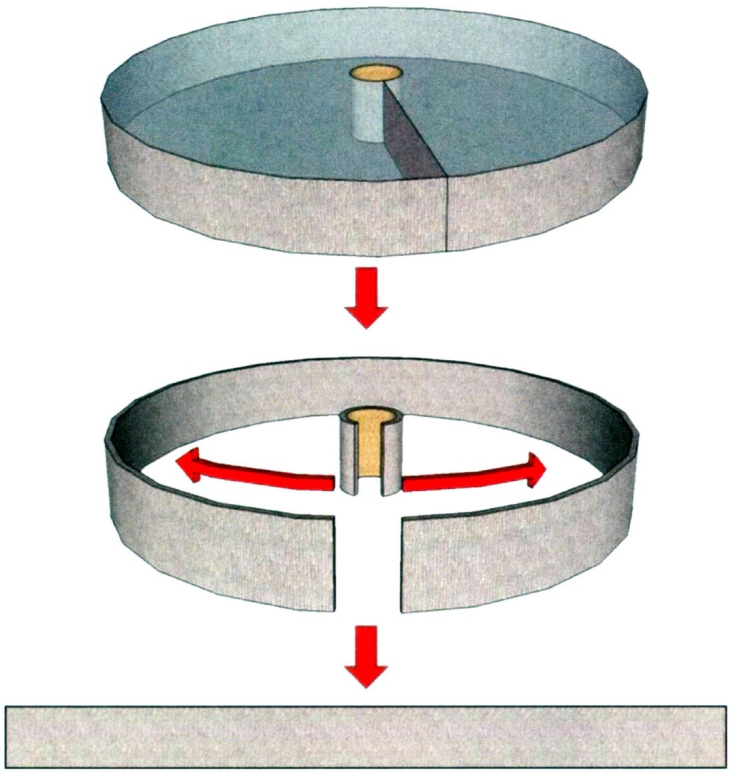


Figure 83 - Unwrapping of Exterior Boundary to a Rectangular Surface

The unwrapped rectangle will have a height h and a length equal to the circumference of the reservoir ($2\pi r_e$). All streamlines start at this face and proceed inwards towards the well as a result of the differential pressure in that direction. To establish the starting points of the streamlines the rectangle may be discretized into a grid composed of a series of equally sized triangles, as illustrated in Figure 84. The vertices of the triangles are uses as the starting points for the streamlines, while the triangles themselves are used as the stream tubes for the flow rate calculations to follow.

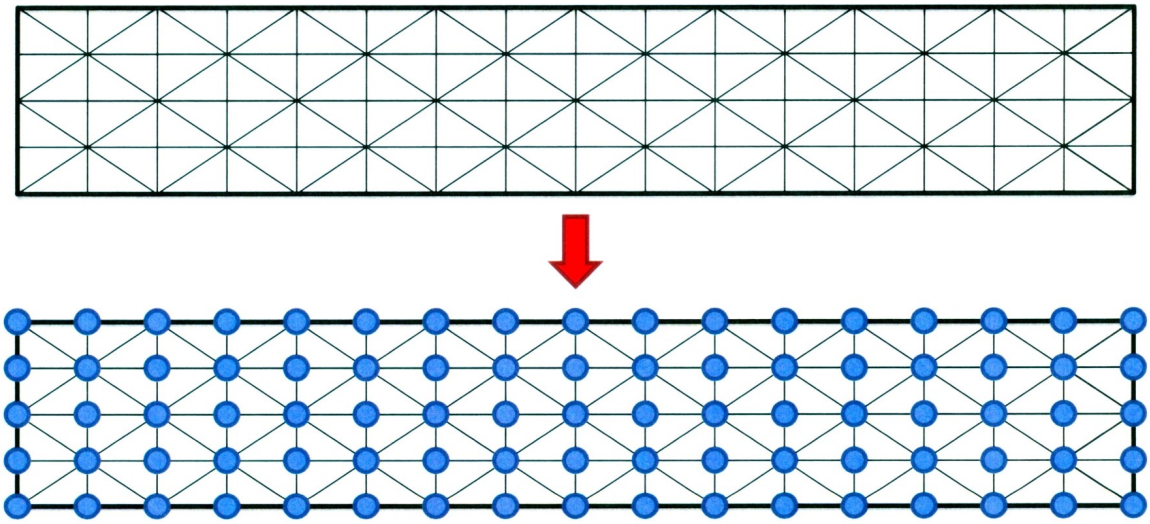


Figure 84 - Triangulation Grid for 3D Mapping

The discretization into triangles may be performed in a number of ways. For tracking consistency it is best to ensure the streamlines start in a rectangular base pattern, forcing each rectangle between four streamlines to consist of two triangular stream tubes. Inputs are required for the number of radial points and the number of layers of points. The spacing and cylindrical coordinates can then be determined based upon these inputs.

Once the starting points for each streamline are established the streamline modeling process can be performed, as described in Section 5.3. The mapping of all streamlines is

carried out in the normal fashion, recording coordinates for each grid face intersection in three dimensions.

Once the streamline mapping is completed the flow rate procedure may begin. Two key principles allow for the tracking of streamlines in a cylindrical reservoir geometry; streamlines can never cross and they will always progress inwards as a result of the positive differential pressure between the external boundary and the well.

In the streamlines modeling process all face boundary intersections are recorded when a streamline comes into contact with a face. For the purpose of these calculations only the points intersecting the radial faces are required, reducing the data set for each streamline to the same number of recorded coordinates. At the radial face intersections the streamlines may again be connected to form a triangle, likely of a different shape and on a different angle as the initial triangle. The two triangles for subsequent radial faces may be connected at each vertex to form a segment of a stream tube. This is illustrated in Figure 85.

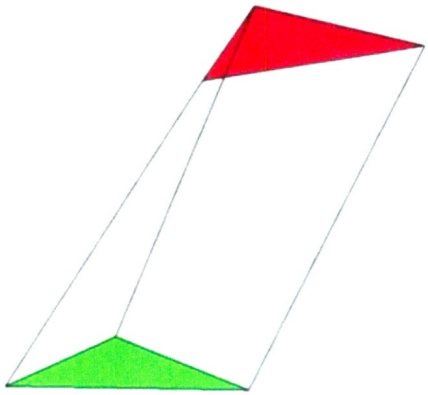


Figure 85 - Triangular Stream Tube Segment

The challenge for the remaining aspects of the flow rate calculation procedure lies in determining the volume occupied within these stream tube segments. To determine this, various elements of vector algebra are required.

As discussed in Chapter 3, the area of a parallelogram formed by two vectors can be determined by calculating the magnitude of the cross product between them, as follows:

$$A_{parallelogram} = |F \times G|$$

$$A_{parallelogram} = |F||G| \cos\theta$$

In the above, F and G are vectors formed by lines connecting three points, with θ representing the angle between them within the plane of the vectors. These three points represent the triangular stream tube intersections with a radial face. The area of a triangle, formed by a third vector H, is simply half the area of the parallelogram, or:

$$A_{triangle} = \frac{1}{2}|F||G| \cos\theta$$

A schematic illustration of this area calculation is provided in Figure 86.

To determine a volume a triangle must be extruded to the next radial triangle along one of the three streamlines connecting the two. In the simplest case the two radial triangles would be of the exact same size and parallel to each other. In this scenario the angles between the triangle plane and each of the streamlines would be the same, resulting in the same area being calculated for the stream tube regardless of which streamline vertex is chosen for extrusion. Figure 87 provides an illustration of the volume calculation for a simplified stream tube segment. The resultant volume is one half of a parallelepiped.

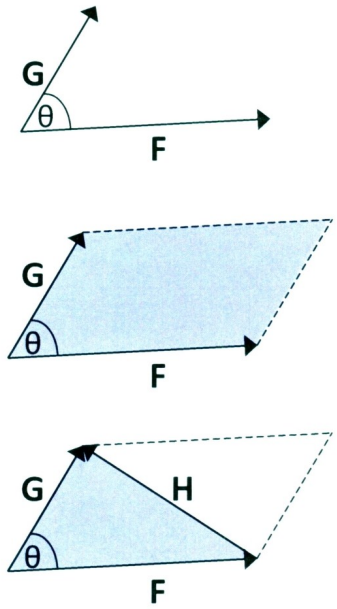


Figure 86 - Triangle Area Calculation

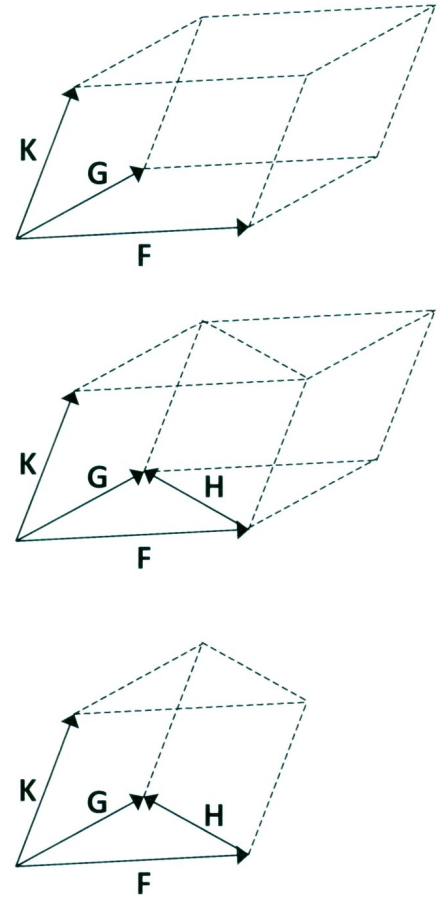


Figure 87 - Parallelepiped Volume Calculation

The volume of the parallelepiped formed by three vectors is the absolute value or magnitude of the scalar triple product of the vectors F, G and K. The scalar triple product is the cross product of the two vectors in the plane (F, G) with the dot product of the vector out of the plane (K). The mathematical expression is:

$$V_{parallelepiped} = |K \cdot (F \times G)|$$

Since the interest is in the extrusion of the triangle formed by vectors G, F and H along the vector K, the volume of the simplified stream tube becomes:

$$V_{half\ parallelepiped} = \frac{1}{2}|K \cdot (F \times G)|$$

The direct scalar triple product procedure outlined above is not sufficient in itself to determine the volume of a realistic stream tube segment. The assumptions of the method include parallel triangles of equal size leading to equal length streamlines at each vertex. In reality, reservoir heterogeneity and the progression to smaller radial faces will cause streamline segments to have different lengths between radial faces. This will force the subsequent triangles in a stream tube to be of a different size than the previous and have different angles at each streamline vertex. The different sizes and angles create six extrusion vectors within a stream tube in which a scalar triple product may be performed. The six extrusion vectors are shown in Figure 88.

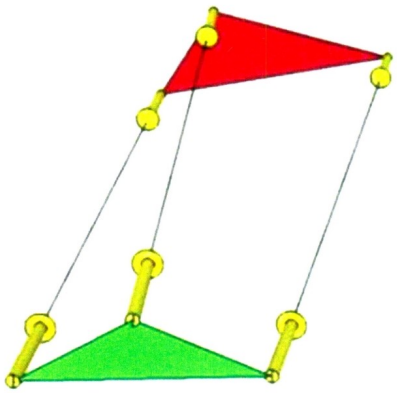


Figure 88 - Triangular Stream Tube Six Vector Extrusion

If the six triple products were to be calculated along each of the six extrusion directions (three vertices for each triangular area) six unique tubes of different volumes would be generated. An illustration of the potential stream tube formations is provided in Figure 89 and Figure 90. Note that translucent vertical baffles are added to enhance the clarity of the figures. The arrows in the figure indicate the extrusion vector for each stream tube segment.

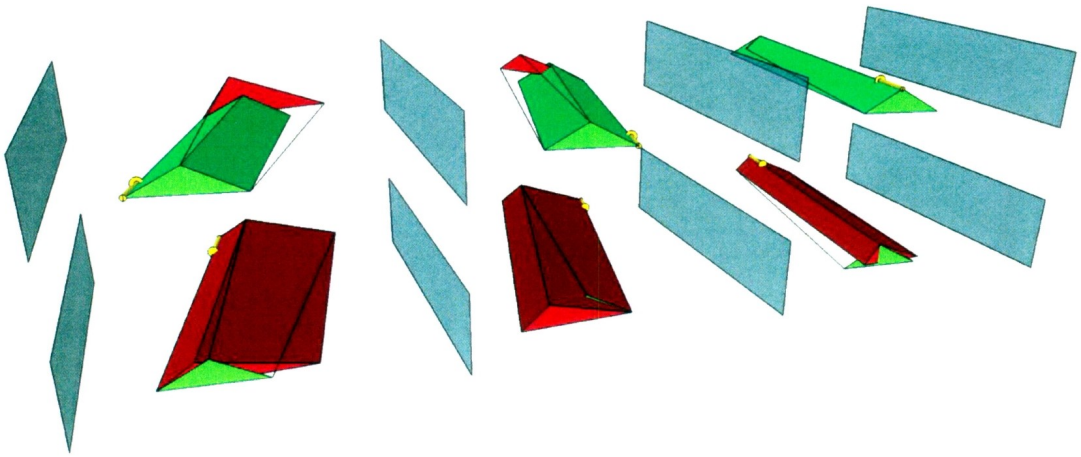


Figure 89 - Six Unique Half Parallelepipeds

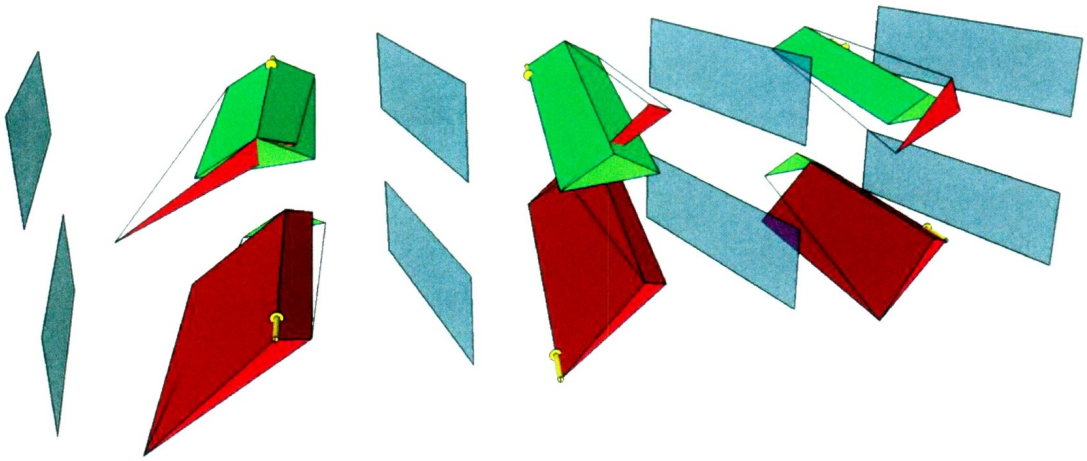


Figure 90 - Six Unique Half Parallelepipeds (reverse direction)

To overcome this issue a simplification must be applied. The area of both triangles can be calculated using the cross products at each respective radial face. Knowing these two areas, an average area for the stream tube cross-section can be determined, as follows:

$$\bar{A} = \frac{1}{2}(A_i + A_{i-1})$$

$$\bar{A} = \frac{1}{2}\left(\frac{1}{2}|F_i||G_i|\cos\theta_i + \frac{1}{2}|F_{i-1}||G_{i-1}|\cos\theta_{i-1}\right)$$

$$\bar{A} = \frac{1}{4}(|F_i||G_i|\cos\theta_i + |F_{i-1}||G_{i-1}|\cos\theta_{i-1})$$

The volume of the stream tube may then be calculated by taking the average length of each streamline segment and multiplying it with the average area, as follows:

$$\bar{x} = \frac{1}{3}(|\vec{x}_1| + |\vec{x}_2| + |\vec{x}_3|)$$

$$V_{stream\ tube\ segment} = \bar{A} \cdot \bar{x}$$

The averaging procedure should be mass conservative. The average area is determined by averaging the exact area at each stream tube segment face. Volume is then calculated by taking the average area and multiplying it by the average streamline segment length. This should be a mass conservative approach, determining an accurate volume for a stream tube segment.

It should be noted that if differences exist, they should be small. In addition, if the averaging procedure for one stream tube segment overestimates, then the adjacent stream tube segment will underestimate. Therefore, the sum of the volume of each stream tube throughout the reservoir domain will equal the total volume of the reservoir.

Knowing the volume of each stream tube segment, the segments along each tube may be summed from the outer boundary to the inner boundary to determine the total volume of the stream tube. For true volume, porosity must be incorporated. Therefore the bulk volume of a particular stream tube segment must be multiplied by the porosity for the segment. If the porosity value changes within the segment an average must be calculated. The pore volume for the stream tube is then calculated as follows:

$$V_{stream\ tube} = \sum_{i=1}^{n} V_{stream\ tube\ segment} \times \phi_{stream\ tube\ segment}$$

The time of flight along each streamline vertex must be averaged to represent a single time of flight for each stream tube, as follows:

$$\bar{t} = \frac{1}{3}\left(|\vec{t}_1| + |\vec{t}_2| + |\vec{t}_3|\right)$$

The total flow along each stream tube is then:

$$q_{stream\ tube} = \frac{V_{stream\ tube}}{\bar{t}}$$

The total flow rate is then calculated by summing the flow contributions of each stream tube, as follows:

$$Q = \sum_{i=1}^{n} q_{stream\ tube}$$

The method established here represents an approximation to determine flow rate for a three dimensional cylindrical coordinate system. Given a sufficient number of streamlines the method should calculate an accurate value for flow regardless of any complex heterogeneities throughout the reservoir medium. An alternative method for any reservoir and wellbore configuration is discussed in the next section.

5.4.2 Generalized Three Dimensional Flow Rate Calculations

5.4.2.1 Assumptions Involved in the Full Field Three Dimensional Procedure

Streamline modeling is a prominent aspect of several commercial ventures within the petroleum industry. In full field three dimensional streamline models (involving multiple wells and complex grid configurations) the tracking of tube geometries can become complex and numerically exhaustive. Therefore, to simplify the calculation procedure stream tubes are no longer mapped. Instead, a more abstract approach is utilized in which

a streamline is viewed as the center of an individual stream tube of known volume but unknown geometry.

Recall from Chapter 1 a fundamental difference exists between full field reservoir models and near wellbore reservoir models. Simply stated, a near wellbore model focuses on a limited region surrounding a single well, while full field models focus on numerous wells in a much larger reservoir domain. As a result, certain procedures and methodologies which may apply to modeling on a field scale may not apply to modeling on a smaller scale. This may be the case with respect to the flow rate modeling methodology utilized within the current full field streamline models.

In full field models, numerous production and injection wells are incorporated throughout the reservoir domain. At each injection well the injection rate is a known input. This is distinctly different from the stream tube methodology presented in the previous section, in which flow rate is an unknown and is the subject of the calculation procedure.

Recall that wells in traditional models are represented as a point in the center of a large grid block. In full field models, all streamlines will start from an injection block and proceed to producers along paths dictated by the streamline modeling process. It is then assumed that the flow rate along each streamline is equal to the total flow rate divided by the number of streamlines (Thiele et al., 1996). A two dimensional graphic providing an illustration of the concept is provided in Figure 91.

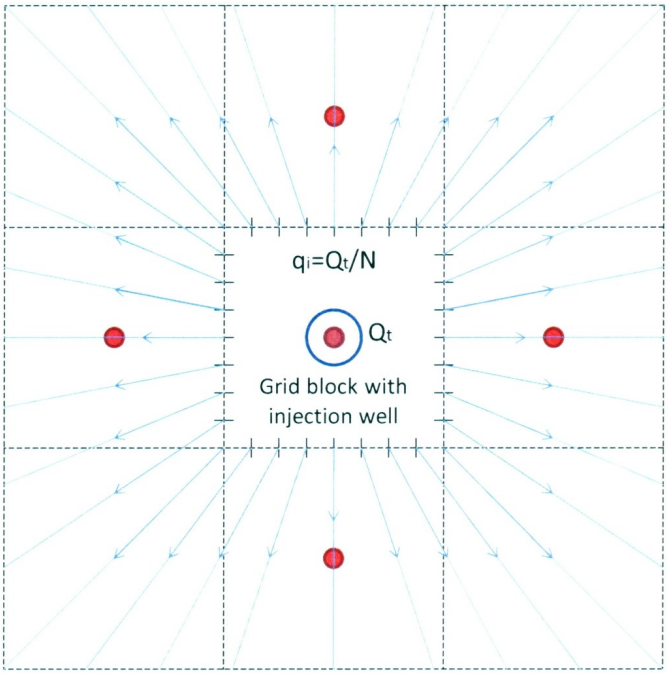


Figure 91 - 3D Flow Rate Assumption

In the figure a slice of a Cartesian grid block is shown. The grid nodes are shown as red circles while the streamlines are shown as blue lines. An injection well is established in the center grid block. Each streamline will now be viewed as the center of an individual stream tube of known volume but unknown geometry. With the total injection flow rate Q_T known, the flow in each stream tube is assumed to be:

$$q_i = Q_T/N$$

In the above N represents the total number of streamlines. Note that the locations of the streamlines are divided equally as required by this assumption.

Knowing the flow rate in each stream tube the volume of the tube can be established based upon the time of flight of the streamlines. In its simplest from, the volume along a tube segment ΔV is:

$$\Delta V = q_i \Delta \tau$$

In the above q_i is the known flow rate along the streamline and $\Delta\tau$ is the time of flight along the segment of interest. If a streamline has a small time of flight it represents a fast moving tube containing little volume. Likewise, a streamline with a large time of flight represents a slow moving tube containing a large volume.

5.4.2.2 Significant Aspects of the Full Field Three Dimensional Procedure

Based upon the assumption highlighted in the previous section, one of the most significant aspects of full field streamline modeling is established. To explain this, one must looks back to the two dimensional case. In two dimensions, using the stream tube approach in which the geometry of the tube is known, a dimensionless length can be established, as follows:

$$x_{D_i} = \frac{\int_0^s \emptyset(\xi) A_i(\xi) d\xi}{\bar{V}_p}$$

where ξ is a coordinate along the streamline, $\emptyset$ is the porosity, A_i is the cross-sectional area of the tube and $\bar{V}_p$ is an arbitrary pore volume used for scaling (Johansen, 1992; Thiele et al., 1996). Likewise, a dimensionless time can be established as follows:

$$t_{D_i} = \frac{Q_t t}{N \bar{V}_p}$$

Knowing these two dimensionless variables it then becomes possible to solve flow and transport problems involving streamlines as a function of x_D and t_D along the streamlines (Johansen, 1992; Thiele et al., 1996).

In three dimensions, using the variables established above, it then becomes possible to model flow and transport without explicitly knowing the tube geometries. Assuming equal flow into each of the stream tubes the ratio of x_D/t_D can be expressed as:

$$\frac{x_D}{t_D} = \frac{\int_0^s \emptyset A_i(\xi) d\xi}{qt} = \frac{1}{t}\int_0^s \frac{\emptyset A_i(\xi) d\xi}{v A_i(\xi)} = \frac{1}{t}\int_0^s \frac{d\xi}{u} = \frac{\tau}{t}$$

In the above $u = v/\phi$ is the Darcy velocity and τ is the time of flight along a streamline from the initial point to a particular point s (Thiele et al., 1996; Johansen, 1992). In knowing that the dimensionless ratio x_D/t_D is equivalent to the ratio τ/t the need to evaluate complex tube geometries is eliminated since the ratio τ/t will only be dependent upon the velocity along a streamline.

The development of this technique not only eliminates the need to evaluate tube geometries, but it also separates the streamlines from the underlying finite difference grids, as well as decouples the effect of heterogeneities from subsequent transport calculations (Datta-Gupta & King, 2007). This is often considered the key step that has allowed streamline simulation to succeed for complex field scale three dimensional problems.

5.4.2.3 Comparison of the Full Field Three Dimensional Assumption to the Stream Tube Approach for Near Wellbore Modeling

The continuation of the streamline modeling process in the evaluation of transport phenomena, temporal updates, gravity, compressibility, history matching and the like is beyond the scope of this thesis. However, the evaluation of the underlying assumption of equal flow in each stream tube can be investigated for its applicability to a near wellbore scale model.

As explained previously, the three dimensional approach involving stream tube calculations is not programmed and, therefore, cannot be investigated. As a substitute, the assumption is tested for the two dimensional approach outlined in Chapter 3. To test the assumption, a generalized case involving eight equally spread blocks of low permeability is analyzed. According to the assumption, the flow rate for each stream tube should be equal. In theory, this may be possible. Streamlines coming near or into contact with low permeability regions tend to spread themselves apart. The increase in volume between the streamlines is counteracted by an increase in the time of flight along the streamlines, and thereby may cancel each other out, generating stream tubes with equal flow rate.

The specific details of the case under investigation are provided in Table 26.

As shown in the table, the number of streamlines per exterior tangential block is three. With 120 tangential blocks in the case, a total of 360 streamlines are generated. This translates into 360 stream tubes between streamlines. Note that porosity is assumed equal throughout the reservoir for ease of calculation. The permeability distribution in the reservoir is provided in Figure 92.

Table 26 – Case D: Parameter Details

Case D	Eight block of low permeability to test the equal stream tube flow rate assumption used in three dimensional streamline modeling			
Factor	**Symbol**	**Name**	**Actual Units**	**Value**
A	r_w	Wellbore Radius	m	0.05
B	r_e	External Radius	m	50
C	N	Radial Blocks		50
D	M	Tangential Blocks		120
E	p_{wf}	Wellbore Pressure	bar	280
F	p_e	External Pressure	bar	300
G	k	Bulk Permeability	Darcy	1
H	h	Reservoir Thickness	m	5
I	μ	Oil Viscosity	cP	1
J		Number of Blocks with Low Permeability		8
K		Block Permeability	Darcy	0.25
L		No. Streamlines Per Block		3

As can be seen in the figure, eight blocks of low permeability are arranged equally throughout the reservoir. The start and end radii of the blocks are approximately 10 m and 20 m. The streamline distribution is provided in Figure 93 and Figure 94.

The streamline models show that streamlines will tend to veer away from the low permeability sectors, entering the regions of high permeability along their path towards the wellbore. The area calculation may now be invoked to determine the flow rate associated with each stream tube.

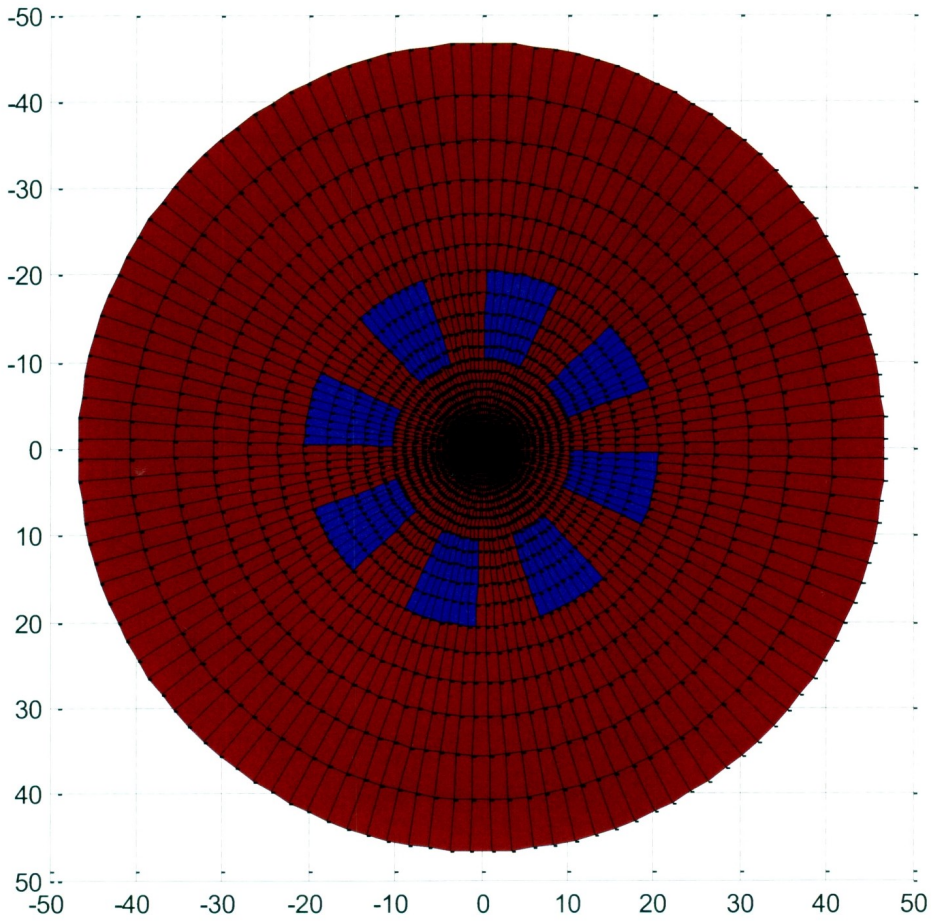


Figure 92 - Case D: Permeability Distribution

Note that the streamlines closest to the low permeability sectors tend to spread apart, creating larger volumes. Also note that clear breaking points are evident in the figures, in which streamlines tend to move left or right of the heterogeneity. In between these streamlines a stream tube of substantial volume is generated.

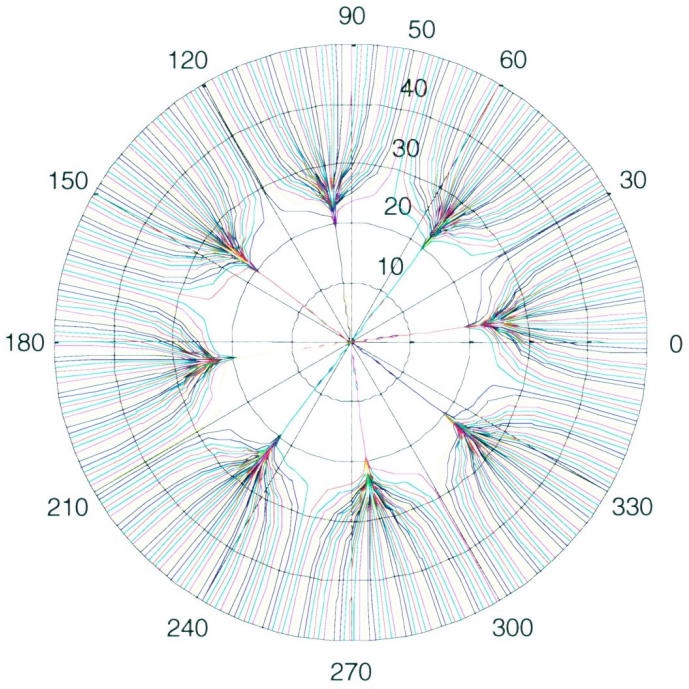


Figure 93 - Case D: Streamline Distribution

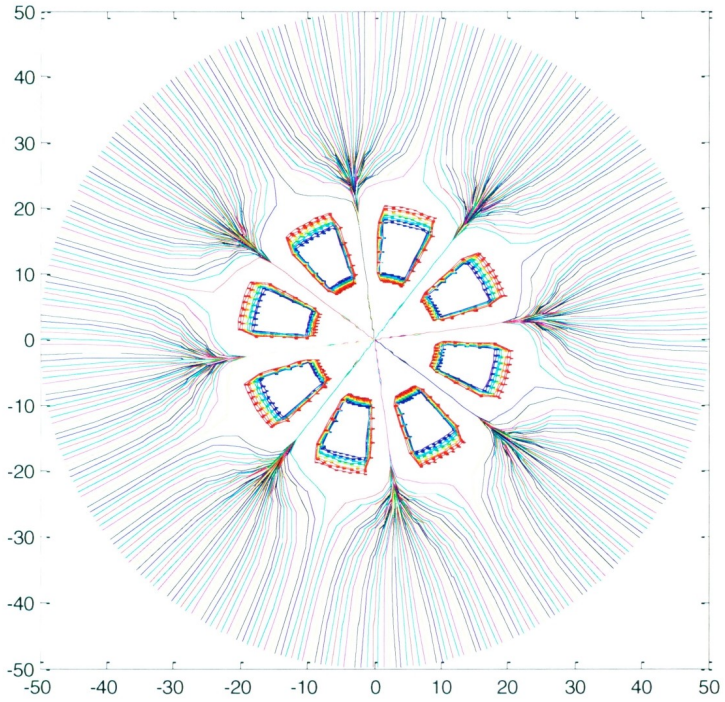


Figure 94 - Case D: Streamline and Permeability Distribution

The flow rate and time of flight associated with each stream tube is plotted in Figure 95.

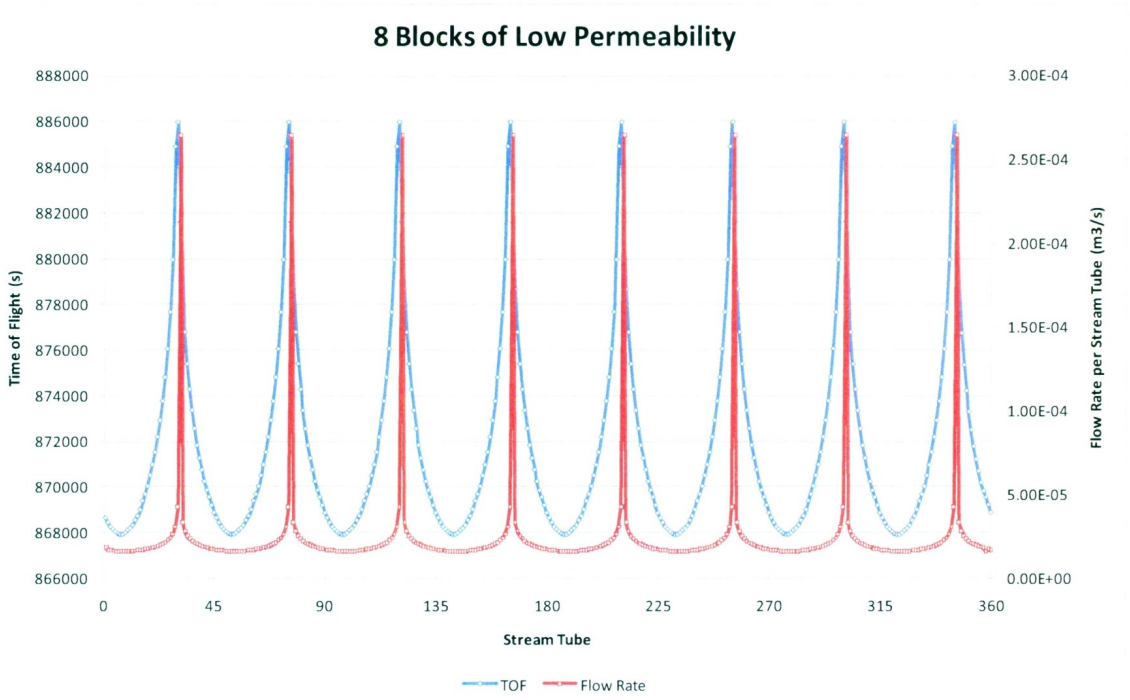


Figure 95 - Case D: Flow Rate and Time of Flight per Stream Tube

As can be seen in the plot, flow rate is not constant throughout each stream tube. While the time of flight increases for tubes nearing low permeability heterogeneities, the increase in volume has a more substantial impact, resulting in larger flow rates. The increase in volume is plotted in Figure 96.

This conclusion may be as a result of discretization errors due to the number or streamlines chosen. Large spacing between streamlines may lead to larger discretization errors. Simply, for stream tubes with large volumes, the average time of flight for the bounding streamlines may not be a good representative for the true average time of flight within the stream tube. An increase in this time of flight will significantly reduce the

peaks in flow rate seen in the figure above. To check this hypothesis, cases with more streamlines are investigated.

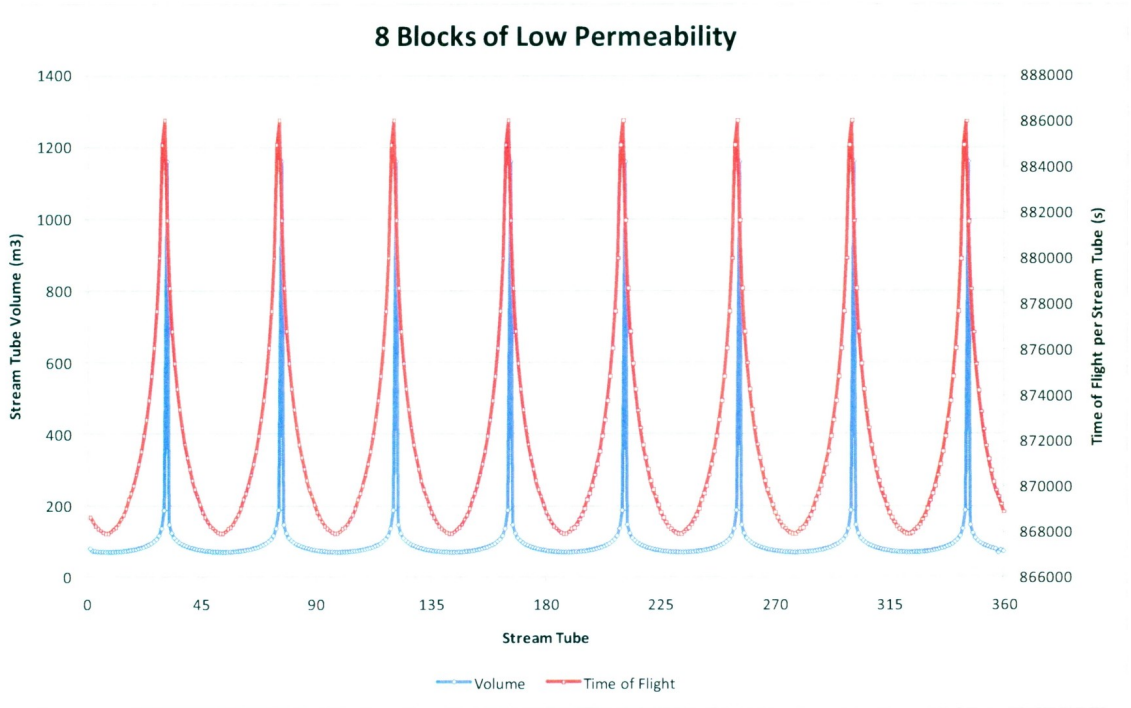


Figure 96 - Case D: Volume and Time of Flight per Stream Tube

For an initial assessment of the effect of the number of streamlines on the flow rate distribution, the same case is run with the exception of having 100 streamlines per tangential block, creating 12 000 streamlines. Polar and Cartesian plots are provided in Figure 97 and Figure 98.

The figures show that regardless of the number of streamlines, the general behavior will remain the same. The streamlines will veer from the low permeability areas to areas of higher permeability as a result of the favorable pressure distribution. As the numbers of streamlines are increased, more "stray" lines occur in the area of the heterogeneity, but an

infinite amount would be required for a full distribution throughout the low permeability regions.

Within the figures, slightly imperfect symmetry is observed around the low permeability blocks. This is as a result of the starting locations of the streamlines. The streamlines are evenly spaced around the exterior of the reservoir. However, this does not coincide with the locations of the streamlines near the heterogeneous grid blocks. Therefore, for streamlines abject to either side of a heterogeneous block, slight differences in symmetry occur. This is as a result of minute location differences with respect to the location of the streamlines.

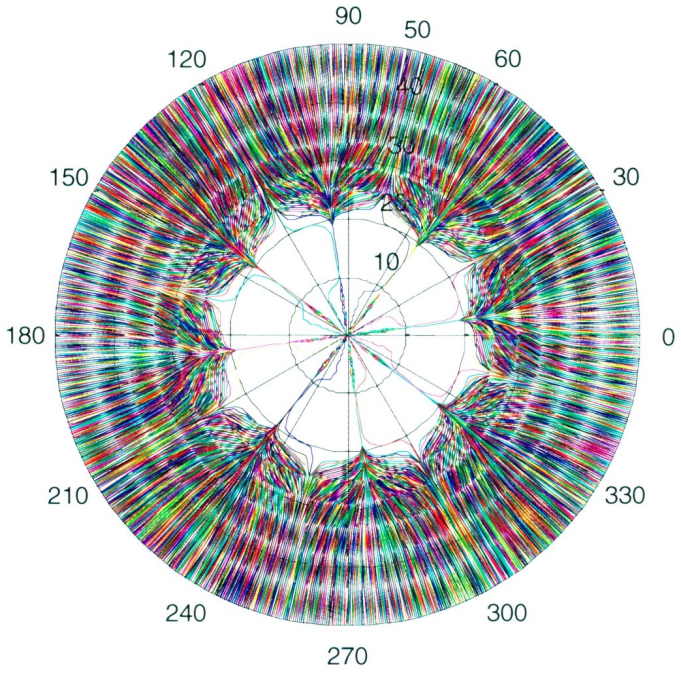


Figure 97 - Case D: Streamline Distribution (12 000 Streamlines)

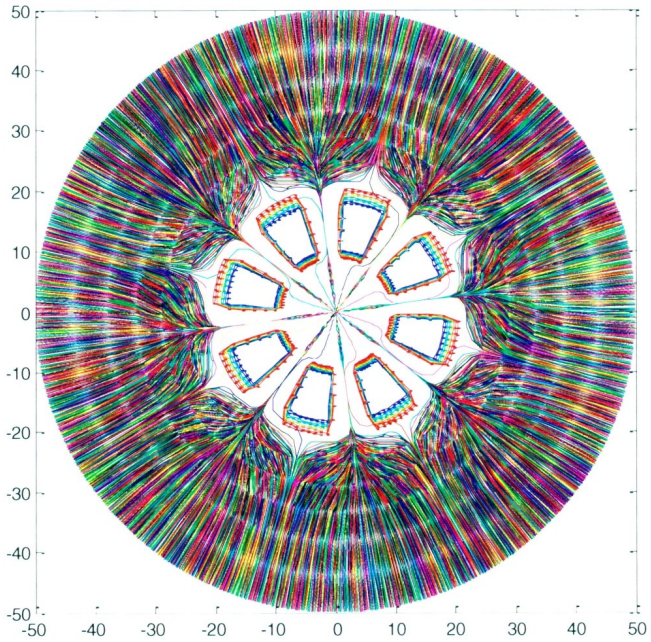


Figure 98 - Case D: Streamline and Permeability Distribution (12 000 Streamlines)

Figure 99 provides a plot of the flow rate per stream tube and the time of flight for the 12 000 streamline case.

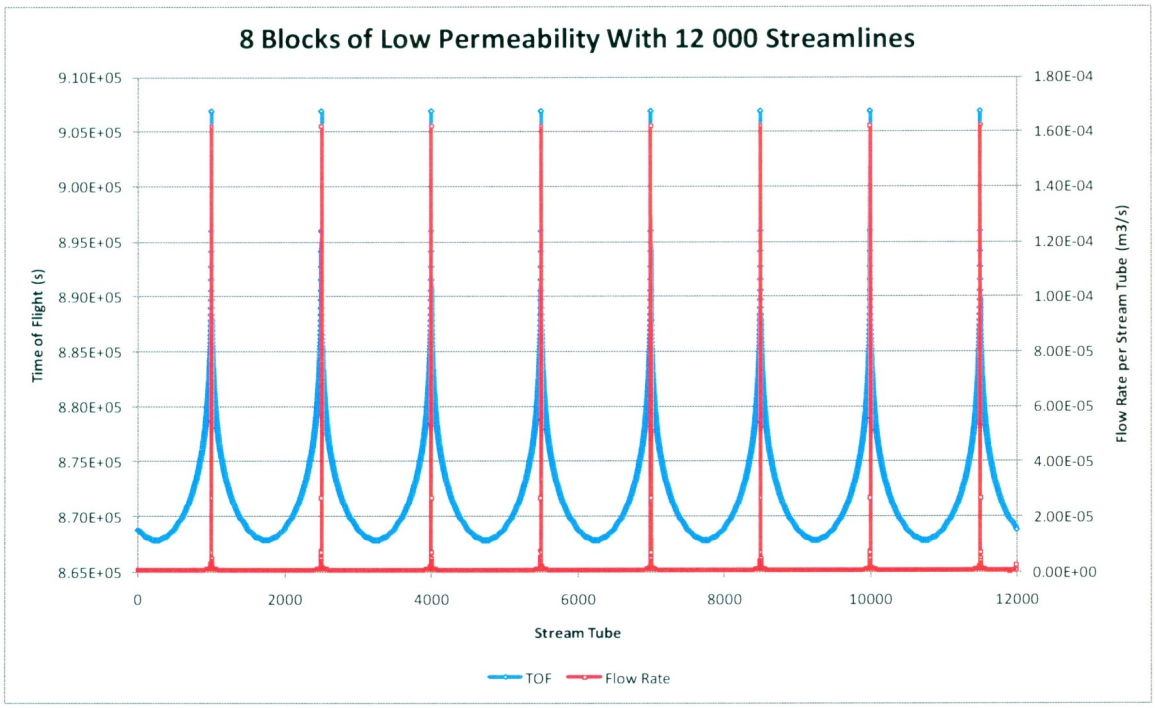


Figure 99 - Case D: Flow Rate and Time of Flight per Stream Tube

To provide a more reasonable example, an additional case is run increasing the block permeabilities from 0.25 Darcy to 0.75 Darcy. This is much closer to the bulk permeability of 1 Darcy, and should allow streamlines to move closer to, and pass through, the heterogeneities. Again, 12 000 streamlines are utilized. Figure 100 provides a view of the streamline and permeability distribution in the reservoir, while Figure 101 provides a magnification of the effects occurring within the area of the heterogeneities.

As can be seen in Figure 101, streamlines tend to spread further apart in the areas of low permeability and come together in the areas of high permeability. As well, unlike the previous case, the streamlines move through the area of low permeability due to the increase from 0.25 D to 0.75 D. The effect of the change on the individual stream tube flow rate distribution is provided in Figure 102.

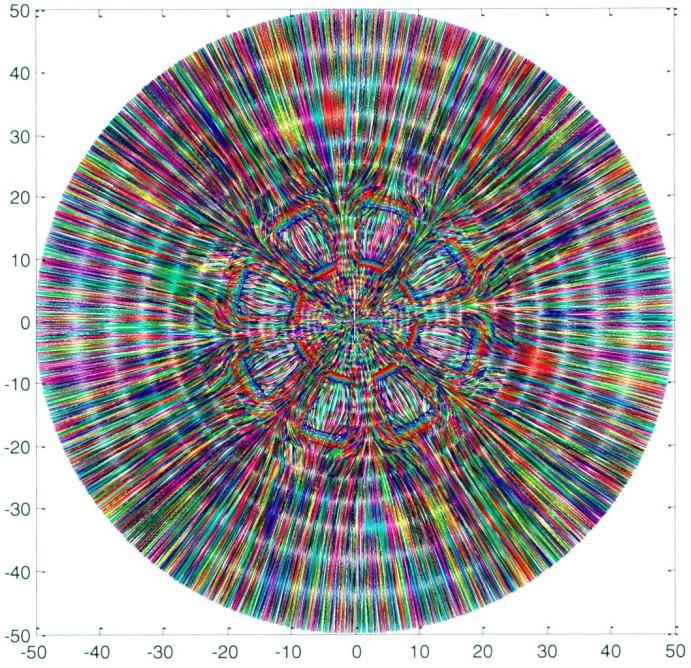


Figure 100 - Streamline Distribution with Increased Block Permeability

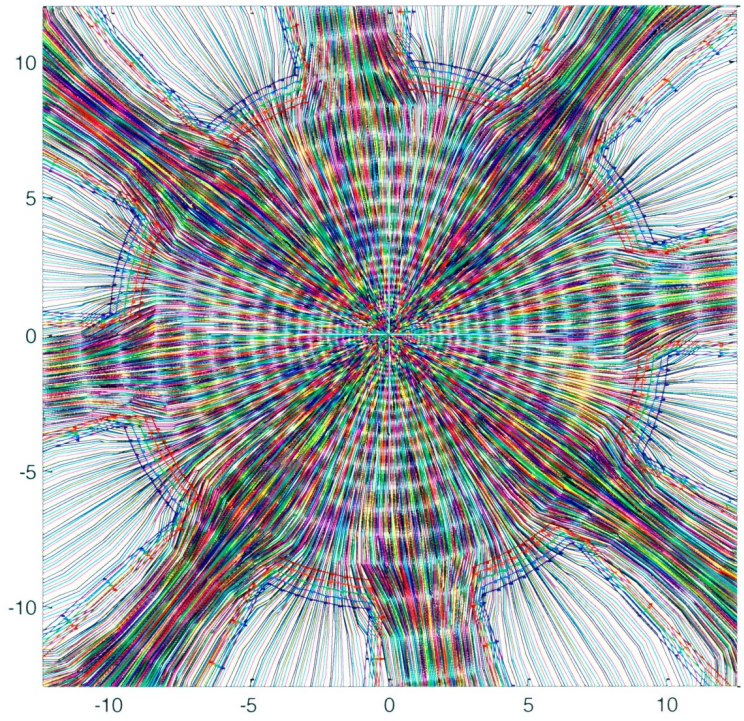


Figure 101 - Close up of Streamlines Through Heterogeneities

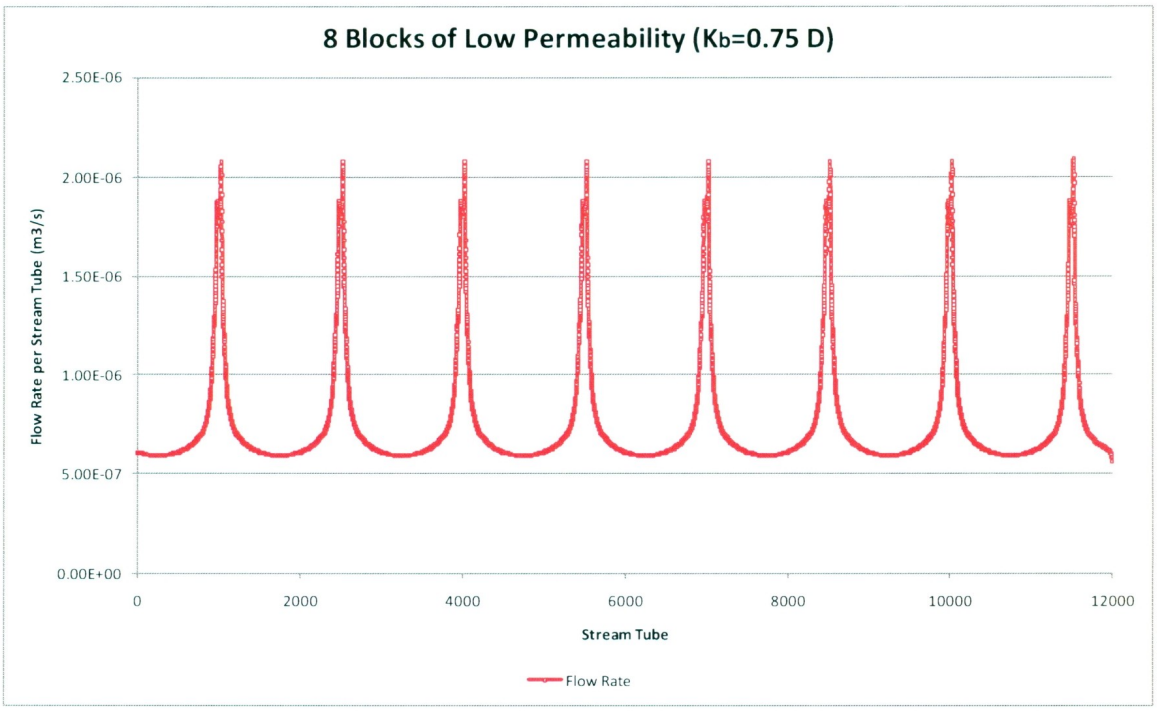


Figure 102 - Flow Rate per Stream Tube with Block Permeability of 0.75 D

As can be seen in the figure, while the magnitude of the peak with respect to the average has diminished, the flow rate throughout each tube is not constant. In areas of low permeability stream tubes will have both larger volumes and larger times of flight. But the increase in volume has proven to be more significant than the increase in time of flight, resulting in a larger flow rate in these regions. This is slightly deceptive, since even though the flow rate is higher, it will take longer to produce as a result of the slower movement of the front. Nevertheless, one can state that in reservoirs in which flow profiles are dominated by heterogeneities, an assumption of equal flow rate in each stream tube may not be a valid approximation. At a minimum, it is not sufficient on a near wellbore scale.

5.4.2.4 Summary of the Stream Tube Equal Flow Rate Assumption

The method presented calculates the volume of a stream tube. In addition, the average time of flight along a stream tube is also calculated by averaging the streamline times of flights which make up the stream tube. The flow rate is then determined by dividing the volume of the stream tube by the average time of flight.

Various authors in full field streamline models use the general assumption of equal flow rate in each stream tube. The justification for this is that stream tubes with larger volumes will have slower times of flight, and stream tubes with smaller volumes will have quicker times of flight. They assume these effects counterbalance each other, generating stream tubes of equal flow rate throughout the reservoir domain.

This work has revealed this assumption to be invalid. While the logic of previous authors is correct with respect to the behavior of stream tube volumes and time of flights, the

effects do not equally balance each other to generate stream tubes of equal flow rate. This research has shown that the stream tubes with larger volumes produce larger flow rates. These stream tubes do have slower time of flights (large travel times) but the increase in volume is more substantial than the increase in time. This results in unequal flow rates throughout the stream tubes in the reservoir medium.

- Chapter 6 -

6.0 Conclusions and Recommendations

6.1 Conclusions

Streamline modeling is a useful technique within petroleum reservoir simulation. Techniques related to streamline modeling have been used in the industry for over 60 years, with many new advances occurring throughout the last 25 years. In the available literature, streamline modeling has proven to be a valuable tool for full field simulation, especially in reservoirs in which flow is convection dominated. Recent research has expanded the capabilities of streamline modeling through the incorporation of various physical and chemical phenomena, including multiphase flow, unsteady state flow, capillary pressure, compressibility and gravity. While many of these advancements require further improvement, they have all helped contribute to the current success of streamline modeling. Today, full field streamline models are used for numerous purposes, including flow visualization, flow surveillance, well allocation, dynamic reservoir characterization, flood management, history matching and general reservoir simulation.

The research presented in thesis outlines the development and usefulness of a near wellbore streamline model. The model developed neglects many of the recent advances in streamline modeling with respect to more complicated physics. Instead, focus is placed upon the fundamentals of the streamline modeling technique and its application to a near wellbore scale. As opposed to focusing on an entire field, the model developed focuses

on a limited region surrounding a single well. In doing so, the ability to model the effect of near wellbore heterogeneities on overall productivity is achieved.

The near wellbore model created in this research is derived for an isotropic heterogeneous medium, therefore permeability may vary with space, but not with direction. The model is created for a vertical well in a cylindrical reservoir. As discussed in the next section, significant novelties of the approach include modeling from a first principles perspective in polar coordinates and logarithmic grid refinement. It is these two alterations that are responsible for the success of the near wellbore streamline model, allowing for accurate modeling of fine resolution permeability changes in the near wellbore vicinity.

The usefulness of the modeling technique presented in this thesis is most evident through the modeling of various completion strategies. Three case studies are performed in this thesis, showing the effect of various near wellbore related properties on the productivity of a well. Two completions are studied in the case studies; a cased and perforated completion, as well as an open-hole completion with filter cake buildup. Through the use of Design of Experiments and Response Surface Methodologies the behavior of the responses within each case study are observed and analyzed.

Case Study 1 performs a 2^{6-1} fractional factorial design on a cased and perforated well completion. The results indicate that the significant factors influencing productivity are the number of perforations, perforation permeability, perforation length, damage zone radius and damage zone permeability. In addition, two interactions are also significant; the interaction between damage zone radius and damage zone permeability as well as the

interaction between perforation length and perforation permeability. The presence of curvature within the response is also detected.

Case Study 2 expands upon the results of Case Study 1, using an IV-optimal design to determine the nature of the responses as a result of changes in perforation characteristics. The results indicate all three factors under study prove significant to the model behavior. The precise nature of the influence of each factor is slightly dependent upon the response analyzed. For flow rate, the number of perforations results in linear behavior, perforation length results in quadratic behavior and perforation permeability results in cubic behavior. Strong interactions are also observed. For skin, a slight increase in model complexity is required. While the number of perforations remains linear and perforation length remains quadratic in nature, the perforation permeability requires quartic behavior. Several additional interactions are also significant within the skin response.

The most significant conclusion drawn from Case Studies 1 and 2 relates to the strong interaction between perforation permeability and perforation length. When perforation permeability is low, changes in perforation length do not result in increased productivity. However, as perforation permeability increases, the effect of perforation length becomes more substantial. Therefore, to provide maximum completion effectiveness, high perforation permeability must be targeted as the main factor of optimization. In turn, this will lead to greater productivity as a result of perforation length and thereby provide optimal completion performance.

In Case Study 3 an IV-optimal design is used to determine the effect of various filter cake properties on the productivity of an open-hole completion. The results indicate only the

angle of coverage has a significant impact on the productivity of a well. This results in a single variable quadratic response for both flow rate and skin. Filter cake thickness and filter cake permeability prove to be insignificant to the response over the range chosen at the desired significance level.

The near wellbore streamline modeling technique established in this research has proven to be valuable for various forms of fine scale completion modeling. Logarithmic grid refinement results in an extremely fine scale grid surrounding the wellbore in the vicinity in which the most significant pressure drop occurs. In traditional finite difference models this would be unfeasible due to the computational burden given the number of grid blocks. However, streamline modeling techniques are noted for their speed with respect to computational efficiency. Therefore, the fine grid scale is not an issue, allowing for detailed modeling within the near wellbore region.

Within the industry this technique has the potential to be utilized for numerous purposes. Future use may allow operators to determine optimal completion strategies, optimal work-over plans, dynamic skin calculation and numerous other applications. At the current level, the numerous assumptions within the model limit its industry applicability. This research has validated the near wellbore streamline technique. However, various assumptions must be lifted and additional complexities added before it may be implemented into a state of the art inflow modeling package. Various recommendations for future work are provided in Section 6.3. The recommendations are aimed at means to enhance the capabilities of the near wellbore streamline model, making it an asset for petroleum installations. Nevertheless, this research has proven the validity of the near

wellbore streamline modeling technique, presenting a means to evaluate the effect of near wellbore heterogeneities with respect to their influence on production.

6.2 Novelty of Research

Numerous novel techniques and developments are presented throughout this research. The generalized modern streamline mapping techniques were put forth by Pollock (1988). However, several novel alterations are performed in this research that allow for detailed streamline modeling in the near wellbore region of an oil well.

First, this research presents the first successful and fully functional use of streamline modeling in the near wellbore region. Nowhere in the published literature has streamline modeling been applied to this vicinity. Streamline modeling in this vicinity is a significant development; it allows for the detailed understanding of how near wellbore heterogeneities affect the pressure distribution, streamline distribution and overall productivity of a well. This can be of great significance in the petroleum industry, allowing for effective decision making through effective completion and work-over design to maximize productivity of individual wells.

Streamline modeling within the near wellbore region is achieved through a first principles derivation for an isotropic heterogeneous medium in polar coordinates. The derivation in itself is novel, showing how the permeability matrix for the general case of an anisotropic heterogeneous media cannot be diagonal when modeling in polar coordinates, even when the principle permeability directions align with the Cartesian grid directions. This is an error made within several sources in the published literature that should be corrected for proper modeling of the Laplace pressure solution. Within this research, proper modeling in polar coordinates is essential to the success of the near wellbore modeling application.

One of the most significant elements contributing to the modeling success is logarithmic grid refinement. As opposed to the traditional equally spaced grid distribution, in this research the grid in the radial direction is spaced such that equal pressure drops occur between adjacent nodes for the simplified case of a homogenous reservoir. Pressure drop in the radial direction is known to decrease logarithmically due to the substantial accumulation of fluids in a diminishing area. Therefore, by applying logarithmic grid spacing, the grid will have a low density in the distant regions of the reservoir domain, but will increase in density as it approaches the wellbore. This results in an extremely fine resolution grid in the very near wellbore vicinity; the region in which the most significant pressure loss occurs. This allows for the modeling of fine scale permeability changes within the vicinity, determining their impact on the productivity of a well. This is a significant development for industry application. It allows for the use of fine scale completion modeling, down to the millimeter scale for cases which need such accuracy. This is an important aspect of this research and is validated through the case studies investigated.

Another novel aspect of the near wellbore streamline modeling technique presented in this thesis relates to the velocity interpolation. In most streamline models, once face velocities are determined, the velocities within the grid blocks are determined through linear interpolation. However, in the radial direction velocity is known to increase inversely proportional to radius. Therefore, as opposed to linear interpolation in the radial direction, this research interpolates in a manner inversely proportional to radius. This allows for an improved velocity distribution, resulting in a reduction in discretization errors.

Through the novel techniques described above the application of near wellbore streamline modeling to evaluate various completion strategies is achieved. In various case studies the model is used to represent a cased and perforated well completion as well as an open-hole completion with filter cake buildup. This is the first time streamline modeling has been used to evaluate completion effectiveness. With implementation of the recommendations for future work, this will have significant potential for industry implementation.

Finally, the last novel technique presented in this research lies within the application of appropriate experiment design and Response Surface Methodologies (RSM) to understand the behavior of the near wellbore streamline model. RSM allows for a simplified understanding of model behavior and is instrumental to the conclusions drawn within each case study. The use of RSM is progressively becoming more common throughout the petroleum industry, and its successful application in this research further validates its usefulness.

6.3 Recommendations for Future Work

As a result of the research performed, several recommendations for future work are provided. The recommendations provided in the following subsections build upon the near wellbore streamline model derived throughout this research, making it more applicable for industry applications.

6.3.1 Extension to Three Dimensional Modeling

Throughout this research modeling is performed in two dimensions, assuming streamlines remain parallel in the vertical plane. This assumption requires the reservoir pay zone to be homogenous in the vertical direction, as well as fully perforated at the wellbore. To eliminate these assumptions and provide a more realistic physical model, a three dimensional method must be developed. The methodology to model streamlines in three dimensions is outlined in Chapter 5. This must be implemented to prove its validity and enhance the capability of the model. In doing so, vertical deviations in streamlines can be achieved. This will result in the ability to model more complex and realistic completions, leading to further industry significance.

6.3.2 Extension to Heterogeneous Anisotropic Media

The modeling performed in this research focuses on an isotropic heterogeneous reservoir, therefore, while flow may vary with space, it cannot vary with direction. To increase the applicability of the model it must be extended to anisotropic heterogeneous media. This is more representative of most reservoirs and will enhance its industry applicability.

Two approaches may be taken to perform this task. The first and most rigorous method is to discretize the complete generalized Laplace equation. The derivation of the full

Laplace equation is provided in Chapter 3 (two dimensions) and Chapter 5 (three dimensions), prior to making the isotropic assumption. Neglecting the assumption will result in a more complicated, but more complete, discretization problem. This may be solved in the same general manner as outlined in this thesis, with only a slight increase in complexity due to anisotropic conditions.

A second approach to this problem lies in is the use of the solution to the isotropic case in addition to an anisotropic transform. In using an anisotropic transform an equivalent permeability may be generated for each grid block, allowing for the transformed pressure solution to be generated utilizing the isotropic approach. Once generated, the anisotropic transform can be utilized to map the true pressure distribution throughout the reservoir. This allows the complicated case to be solved using the solution to the simpler scenario. The applicability of this transform, as well as a comparison to the rigorous solution, is recommended to be the subject of future research.

6.2.3 Rigorous Solution to Casing Representation

In Case Studies 1 and 2 the near wellbore streamline model is used to represent a cased and perforated well completion. As discussed in Section 4.3.2, the casing is represented at the internal wellbore ring as a region of extremely low permeability. This is meant to influence the pressure solution such that streamlines will veer away from the casing towards the nearest perforation. However, in some instances streamlines will hit the casing ring as a result of numerical errors and proceed through. This represents an unrealistic scenario and is dealt with via a programming alteration.

To properly handle these issue no-flow boundaries should be invoked. A no-flow boundary would be represented by implementing zero differential pressure between the casing nodes and the reservoir nodes in areas in which no perforations exist. This must be implemented in the discretization of the Laplace equation, prior to calculating the pressure solution. In doing so, a true representation of streamline behavior will be achieved without need for code alteration. In modeling more realistic well completions this is a recommendation which must be invoked to achieve industry acceptance.

If a less complex solution is desired, a harmonic mean can be used for radial upscaling along the interior casing ring. Harmonic upscaling is commonly used for radial upscaling in heat transfer conductivity codes with great success. This may help mitigate the error associated with streamlines hitting the casing, forcing more curvature.

6.2.4 Incorporation of More Complex Physics

In this research numerous simplifying assumptions are made. These include single phase flow, steady state flow, incompressible fluid and negligible capillary pressure and gravity effects. Over the last 25 years significant progress has been made in applying these physical phenomena to streamline models. Further research with respect to near wellbore streamline modeling should focus on incorporating the advances made by various authors on full field Cartesian coordinate streamline models. In doing so, the accuracy of the model will be increased, making it a fully functional tool for petroleum industry applications.

6.2.5 Enhanced Simulation Well Connection Factor Design

Currently simple techniques are used to connect remote reservoir grid blocks of traditional full field finite difference models to wells located within the blocks. Often, simple productivity index values are used as connection factors between the grid blocks and the well. The approach to generating these values, as well as their accuracy, is often in error. Near wellbore streamline modeling may potentially be used in determining a more accurate approach to connect remote reservoir blocks to wells. This will likely require local grid refinement in grid blocks containing wells, but should not result in a significant increase in computational burden due to the speed of streamline models. As a result, this should be the subject of further research, with the aim of providing significant improvement in full field connection factor design.

6.2.6 Evaluation of New Streamline Modeling Techniques

Potential improvement may be made to the streamline modeling techniques presented in this research. Two errors may occur as a result of the current process. The first error relates to velocity interpolation. In the tangential direction a linear interpolation is utilized, while in the radial direction interpolation is performed inversely proportional to radius. In both cases the two face velocities of a single grid block are used for the interpolation. However, these interpolations may not suffice in accuracy to represent the true velocity behavior. One means to improve upon this may be to investigate the potential of incorporating more known velocities (surrounding grid faces) to help determine the proper order of interpolation to utilize within a block. This investigation may be the subject of future research.

An additional down fall of the Pollock streamline tracking approach relates to the nature of the tracking process itself. Once a particle enters a grid block it must progressively move forward. It cannot exit from the same face as the entrance face in a different location. However, certain circumstances may require this movement, making the current streamline tracking methodology insufficient. Fully analytical methods capable of accurately modeling streamline movement within grid blocks may solve this issue, and as such should be the subject of future research.

6.2.7 Evaluation of Non-Darcy Flow Effects

Section 4.7 of this thesis determined the rate at which flow in the near wellbore region at the face of a fully perforated completion no longer behaves in a laminar manner. Using various assumptions a value of just over $10\,000\ bbl/d$ is calculated. One of the most essential assumptions in this calculation is laminar flow ending at a Reynolds number of 10. Various sources in published porous media literature state laminar flow may end at even smaller values, with some sources suggesting a Reynolds number as low as 1 to be an accurate value. This may significantly impact the primary assumption of streamline modeling; laminar flow. Therefore future research should attempt to establish a credible value at which laminar flow terminates in porous reservoir media. In addition, if the assumption of laminar flow fails for high productivity oil wells, research should be performed on the incorporation of non-Darcy flow effects in near wellbore streamline models.

6.2.8 Dynamic Skin Calculation in Full Scale Reservoir Simulation

With the implementation of several of the recommendations listed previously, the near wellbore streamline model may be extremely useful within the petroleum industry. One

of its greatest potential benefits may lie in its incorporation into full scale reservoir simulators. As such, it may be utilized to dynamically calculate skin with each time step during reservoir simulation. This cannot be achieved in modern reservoir simulators and would represent a significant improvement in terms of individual well production forecasting and work-over evaluation. This should be the subject of future research and the ultimate goal of near wellbore streamline modeling.

Bibliography

Ahmadloo, F., Asghari, K., & Araghi, M. N. (2009). Heavy Oil Viscosity Prediction Using Surface Response Methology. *Canadian International Petroleum Conference (CIPC).* Calgary: Petroleum Society.

Amudo, C., Graf, T., Harris, N., Dandekar, R., Amor, B., & May, R. (2008). Experimental Design and Response Surface Models as a Basis for Stochastic History Match - A Niger Delta Experience. *International Petroleum Technology Conference.* Kuala Lumpur: International Petroleum Technology Conference.

Batycky, R. P., Blunt, M. J., & Thiele, M. R. (1997). A 3D Field-Scale Streamline-Based Reservoir Simulator. *SPE Reservoir Engineering* , 246-254.

Bear, J. (1972). *Dynamics of Flow in Porous Media.* New York: Drover Publications.

Bear, J. (1988). *Dynamics of Flow in Porous Media.* New York: Drover Publications.

Bobeck, P. (2004). *The Public Fountains of the City of Dijon, Henry Darcy, 1856, Translated from French.* Austin, Texas: Geotech.

Bratvedt, F., Bratvedt, K., Buchholz, C. F., Holden, L., Holden, H., & Risebro, N. H. (1992). A New Front-Tracking Method for Reservoir Simulation. *SPE Reservoir Engineering* , 107-116.

Bratvedt, F., Gimse, T., & Tegnander, C. (1996). Streamline Computations for Porous Media Flow Including Gravity. *Transport in Porous Media* , 63-78.

Burton, R. C. (1999, December). Use of Perforation-Tunnel Permeability To Assess Cased Hole Gravelpack Performance. *SPE Drilling and Completions* , 235-239.

Carreras, P. E., Johnson, S. G., & Turner, S. E. (2006). Tahiti: Assessment of Uncertainty in Deepwater Reservoir Using Design of Experiments. *SPE Annual Technical Conference and Exhibition.* San Antonio: Society of Petroleum Engineers.

Carreras, P. E., Turner, S. E., & Wilkinson, G. T. (2006). Tahiti: Development Strategy Assessment Using Design of Experiments and Response Surface Methods. *SPE Westerm Regional/AAPG Paxific Section/GSA Cordilleran Section Joint Meeting.* Anchorage: Society of Petroleum Engineers.

Cerasi, P., Larsen, I., Stenebraten, J., & Sonstebo, E. (2006). Scratch Testing of Drilling Mud Filter Cakes. *SPE Europec/EAGE Annual Conference and Exhibition.* Vienna: Society of Petroleum Engineers.

Cheng, H., Oyerinde, D., Datta-Gupta, A., & Milliken, W. (2007, December). Compressible Streamlines and Three-Phase History Matching. *SPE Journal* , 475-485.

Cornell, J. (1990). *How to Apply Response Surface Methodology. The ASQC Basic References in Quality Control: Statistical Techniques* (Vol. 8). Wisconsin: ASQC.

Dake, L. P. (2001). *Developments in petroleum Sciecne 8: Fundamentals of Reservoir Engineering.* New York: Elsevier.

Darcy, H. (1856). *Les Fontaines Publiques de la ville de Dijon.* Paris: Dalmont.

Datta-Gupta, A. (2000). Streamline Simulation: A Technology Update. *Jounral of Petroleum Technology* , 68-74.

Datta-Gupta, A., & King, M. J. (1995). A Semianalytic Approach to Tracer Flow Modeling in Hetrogenous Permeable Media. *Advances in Water Resources* , 9-14.

Datta-Gupta, A., & King, M. J. (2007). *Streamline Simulation: Theory and Practice.* Richardson, TX: Society of Petroleum Engineers.

Dupuit, J. (1857). Mouvement de l'eau a tavers le terranis permeables. *C. R. Hebd. Seances Acad. Sci.* , 92-96.

Emanuel, A. S., & Milliken, W. J. (1997). Application of Streamtube Techniques to Full-Field Waterflood Simulation. *SPE Reservoir Engineering* , 211-217.

Emanuel, A. S., Alameda, G. K., Behrens, R. A., & Hewett, T. A. (1989). Reservoir Performance Prediction Methods Based on Fractal Geostatistics. *SPE Reservoir Engineering* , 311-318.

Faidi, S. A., Ponting, D. K., & Eagling, T. L. (1996). Experimental Design in Interactive Reservoir Simulation. *Petroleum Computer Conference* (pp. 189-194). Dallas, TX: Society of Petroleum Engineers.

Fay, C. H., & Prats, M. (1951). The Application of Numerical Methods to Cycling and Flooding Problems. *3rd World Petroleum Congress.* The Hague: Society of Petroleum Engineers.

Fisher, R. A. (1918). The Correlation Between Relatives on the Supposition of Mendelian Inheritance. *Transactions of the Royal Society of Edinburgh , 52*, 399-433.

Glimm, J., Isaacson, E., Marchesin, D., & McBryan, O. (1981). Front Tracking in Hyperbolic Systems. *Advances in Applied Mathematics* , 91-119.

Glimm, J., Lindquist, B., McBryan, O. A., Plohr, B., & Yaniv, S. (1983). Front Tracking for Petroleum Reservoir Simulation. *Reservoir Simulation Symposium* (pp. 41-49). San Francisco, CA: Society of Petroleum Engineers.

He, Z., Parikh, H., & Datta-Gupta, A. (2002). Identifying Reservoir Compartmentalization and Flow Barriers Using Primary Production: A Streamline Approach. *SPE Annual Technical Conference and Exhibition.* San Antonio: Society of Petroleum Engineers.

Higgins, R. V., & Leighton, A. J. (1962). *A Computer Method to Calculate Two-Phase Flow in Any Irregularly Bounded Porous Medium.* Bakersfield, CA: Society of Petroleum Engineers.

Hoos, H. (2003). *Space-Filling Designs for Computer Experiments.* Retrieved January 2011, from Computer Science - University of British Columbia: http://www.cs.ubc.ca/~hoos/Courses/Trento-06/module-6.2-slides.pdf

Ibrahim, H., Harrasi, A., Salsman, A., Nunez, A., & Situmorang, H. (2009). Overcoming Near Wellbore Damage Induced Flow Impairment with Improved Perforation Job Design and Execution Methods. *SPE European Formation Damage Conference.* Scheveningen: Society of Petroleum Engineers.

Ingebrigsten, L., Bratvedt, F., & Berge, J. (1999). A Streamline Based Approach to Solution of Three-Phase Flow. *SPE Reservoir Simulation Symposium.* Houston: Society of Petroleum Engineers.

Intelligent Solutions Inc. (2010). *Top Down Intelligent Reservoir Models*. Retrieved December 8, 2010, from http://www.intelligentsolutionsinc.com/TopDownModel.htm

Jimenez, E. A., Datta-Gupta, A., & King, M. (2008). Full Field Streamline Tracing in Complex Faulted Systems with Non-Neighbor Connections. *SPE/DOE Improved Oil Recovery Symposium.* Tulsa: Society of Petroleum Engineers.

JMP. (2005). *JMP Design of Experiments, Release 6.* Cary, NC: SAS Institute Inc.

Johansen, T. E. (2008). *Principles of Reservoir Engineering.* St. John's: Memorial University of Newfoundland.

Johansen, T. E. (2009). *Streamline Modeling in a Radial Geometry.* St. John's: Memorial University of Newfoundland.

Johansen, T. (1992). Shocks and Simple Waves. In S. Skjaeveland, & J. Kleppe, *Recent Advances in Improved Oil Recovery Methods for North Sea Sandstone Reservoirs* (pp. 73-86). Stavanger: SPOR Monograph.

Juanes, R., Lie, K., & Kippe, V. (2004). A Front Tracking Method for Hyperbolic Three-Phase Models. *9th European Conference on the Mathematics of Oil Recovery.* Cannes, France: European Assoication of Geoscientists and Engineers.

Jun, Y., & Shiyi, Z. (2009). New Advance in Numerical Well Testing Through Streamline Simulation. *SPE Asia Pacific Oil and Gas Conference and Exhibition.* Jakarta: Society of Petroleum Engineers.

Kabir, A. H., Salmachi, P., & Salmachi, A. (2009). An Improved Method for Total Mechanical Skin Calculation for Perforated Completions. *SPE EUROPEC/EAGE Annual Conference and Echibition.* Amsterdam: Society of Petroleum Engineers.

Kalla, S., & White, C. D. (2007). Efficient Design of Reservoir Simulation Studies for Development and Optimization. *SPE Reservoir Evaluation and Engineering* , 629-638.

Karakas, M., & Tariq, S. (1991). Semianalytical Productivity Models for Perforated Completions. *SPE Production Engineering* , 73-82.

Kippe, V., Haegland, H., & Lie, K. (2007). A Method to Improve Mass Balance in Streamline Methods. *SPE Reservoir Simulation Symposium.* Houston: Society of Petroleum Engineers.

Kumar, S., Datta-Gupta, A., & Jimenez, E. (2009). Understanding Reservoir Mechanisms Using Phase and Component Streamline Tracing and Visualization. *SPE Annual Technical Conference and Exhibition.* New Orleans: Society of Petroleum Engineers.

Lake, L. W., Johnston, J. R., & Stegemeier, G. L. (1981). Simulation and Performance Prediction of a Large-Scale Surfactant/Polymer Project. *SPEJ* , 731-739.

LeBlanc, J. L., & Caudle, B. H. (1971). A Streamline Model for Secondary Recovery. *SPEJ , 1*, 7-12.

Liu, J., Parker, D., & Camilleri, D. (1999). A New Particle Tracking Algorithm for Tracer Flow Simulation. *SPE Reservoir Simulation Symposium.* Houston: Society of Petroleum Engineers.

Lohne, A., Han, L., Van der Zwaag, C., Van Velzen, H., Mathisen, A., Twynam, A., et al. (2009). Formation Damage and Well Productivity Simulation. *SPE European Formation Damage Conference.* Scheveningen: Society of Petroleum Engineers.

Lye, L. M. (2009). *ENGI 9516 Course Notes: Design of Experiments.* St. John's: Memorial University of Newfoundland.

Martin, J. C., & Wegner, R. E. (1979). Numerical Solutions of Multiphase, Two-Dimensional Incompressible Flow Using Stream-Tube Relationships. *SPEJ , 19*, 313-323.

Mathews, J. L., Emanuel, A. S., & Edwards, K. A. (1989). Fractal Methods Improve Mitsue Miscible Predictions. *Journal of Petroleum Technology* , 1136-1149.

Matringe, S. F., & Gerritsen, M. G. (2004). On Accurate Tracking of Streamlines. *SPE Annual Technical Conference and Exhibition.* Houston: Society of Petroleum Engineers.

Morel-Seytoux, H. J. (1966). *Unit Mobility Ratio Displacement Calculations for Pattern Floods in Homogeneous Medium.* La Habra, CA: Society of Petroleum Engineers.

Morris, J. P., Lomov, I. N., & Glenn, L. A. (2000). *Simulating Perforation Permeability Damage and Cleanup.* Livermore: U.S. Department of Energy.

Muskat, M. (1937). *The Flow of Homogeneous Fluids Through Porous Media.* Boston: International Human Resources Development Corporation.

National Aeronautics and Space Administration (NASA). (n.d.). *Definition of Streamlines*. Retrieved Jaunary 20, 2011, from Glenn Research Center: http://www.grc.nasa.gov/WWW/K-12/airplane/stream.html

Nelson, P. H. (2009, March). *Pore-throat sizes in sandstones, tight sandstones, and shales.* Retrieved from Geo Science World - American Association of petroleum Geologists: http://aapgbull.geoscienceworld.org/cgi/content/abstract/93/3/329

Nilsen, H., & Lie, K. (2009). Front-Tracking Methods for Use in Streamline Simulation of Compressible Flow. *SPE Reservoir Simulation Symposium.* Woodlands: Society of Petroleum Engineers.

Ochi, J., Detienne, J.-L., Rivet, P., & Lacourie, Y. (1999). Extenral Filter Cake Properties During Injection of Produced Waters. *SPE European Formation Damage Conference.* The Hague: Society of Petroleum Engineers.

O'Neil, P. V. (2003). *Advanced Engineering Mathematics 5th Edition.* Toronto: Brooks/Cole-Thomson Learning.

Osher, S., & Sethian, J. A. (1988). Fronts Propagating with Curvature Dependent Speen: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics , 79*, 12-49.

Peddibhotla, S., Cubillos, H., Datta-Gupta, A., & Wu, C. H. (1996). Rapid Simulation of Multiphase Flow Through Fine-Scale Geostatistical Realizations Using a New, 3D,

Streamline Model: A Field Example. *Petroleum Computer Conference.* Dallas, TX: Society of Petroleum Engineers.

Peddibhotla, S., Datta-Gupta, A., & Xue, G. (1997). Multiphase Streamline Modelinf in Three Dimensions: Further Generalizations and a Field Application. *SPE Reservoir Simulation Symposium.* Dallas, TX: Society of Petroleum Engineers.

Pitoni, E., & Ballard, D. (1999). Changes in Solids Composition of Reservoir Drill in Fluids During Drilling and the Impact on Filter Cake Properties. *SPE European Formation Damage Conference.* The Hague: Society of Petroleum Engineers.

Pitts, G. N., & Crawford, P. B. (1970). Low Areal Sweep Efficiencies in Flooding Heterogeneous Rock. *9th Biennial Symposium, Production Techniques.* Witchita Falls, TX: Society of Petroleum Engineers.

Pollock, D. W. (1988). Semianalytical Computation of Path Lines for Finite-Difference Models. *Ground Water , 26* (6), 743-750.

Risso, F. V., Risso, V. F., & Schiozer, D. J. (2007). Risk Assessment of Oil Fields Using Proxy Models: A Case Study. *8th Canadian International Petroleum Conference.* Calgary: Petroleum Society.

Ritzi Jr., R. W., & Bobeck, P. (2008). Comprehensive Principles of Quantitative Hydrogeology Established by Darcy (1856) and Dupuit (1857). *Water Resources Research , 44* (W10402), 1-14.

Schafer-Perini, A. L., & Wilson, J. L. (1991). Efficient and Accurate Front Tracking for Two-Dimensional Groundwater Flow Models. *Water Resources Research* , 1471-1485.

Scheidegger, A. E. (1960). *The Physics of Flow Through Porous Media.* New York: The MacMillan Company.

Scheidegger, A. E. (1974). *The Physics of Flow Through Porous Media.* New York: The MacMillan Company.

Sethian, J. A. (1996). *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge: Cambridge Monograph on Applied and Computational Mathematics.

Shedid, S. A., & Zekri, A. Y. (2006). An Experimental Approach of Influences of Perforated Length and Fractures on Horizontal Well Productivity. *Journal of Canadian Petroleum Technology , 45*, 43-48.

Sheng, J. J. (2010). Discussion of Permeability Aniostropy Effect in Transformation. *Journal of Canadian Petroleum Technology* , 42-46.

SINTEF. (2010). *MRST - MATLAB Reservoir Simulation Toolbox*. Retrieved December 8, 2010, from MRST / Tutorials / Real-Field Model: II: http://www.sintef.no/Projectweb/MRST/Tutorials/Real-Field-Model-II/

Stat-Ease. (2011). *Design Expert 8.0 Help Guide.* Minneapolis: Stat-Ease.

Thiele, M. R. (2001). Streamline Simulation. *6th International Forum on Reservoir Simulation.* Schloss Fuschl, Austria.

Thiele, M. R., Batycky, R. P., & Fenwick, D. H. (2010). Streamline Simulation for Modern Reservoir-Engineering Workflows. *Journal of Petroleum Technology - Distinguished Author Series* , 64-70.

Thiele, M. R., Batycky, R. P., Blunt, M. J., & Orr Jr., F. M. (1996). Simulating Flow in Heterogeneous Systems Using Streamtubes and Streamlines. *SPE Reservoir Engineering* , 5-12.

Thiele, M. R., Fenwick, D. H., & Batycky, R. P. (2007). Streamline-Assisted History Matching. *9th Internatiional Forum on Reservoir Simulation.* Abu Dhabi: Streamsim Technologies Inc.

Vanegas, J. W., & Cunha, L. B. (2006). Prediction of SAGD Performance Using Response Surface Correlations Developed by Experimental Design Techniques. *7th Canadian International Petroleum Conference.* Calgary: Petroleum Society.

White, C. D., Willis, B. J., Narayanan, K., & Dutton, S. P. (2001). Identifying and Estimating Significant Geological Parameters With Experimental Design. *SPE Journal* , 311-324.

Yildiz, T. (2006). Assessment of Total Skin Factor in Perforated Wells. *SPE Reservoir Evaluation and Engineering* , 62-76.

Zangeneh, N., Azizian, A., Lye, L., & Popescu, R. (2002). Application of Response Surface Methodology in Numerical Geotechnical Analysis. *55th Canadian Society for Geotechnical Conference.* Hamilton: CSG.

List of Appendices

Appendix I - Two Dimensional Isotropic Heterogeneous Near Wellbore Streamline Model MATLAB™ Code

The following table outlines the MATLAB™ code files used to develop the near wellbore streamline model. The files are listed and numbered in alphabetical order. A brief description of what each file is used for is provided in the table. For additional detail please refer to the comments provided within the code. The code is provided in the order shown within the table.

No.	Filename	Description	Page
1	casing_perm_fix	File used to set first two node pressures equal to each other, resulting in a differential pressure of zero and hence no flow in the radial direction. It is applied to areas in which casing is present, forcing flow into the perforations.	261
2	darcy_flow_check	File used to determine the flow rate expected from Darcy's law for radial flow, used for comparison to the flow rate generated from the streamline model.	262
3	distances	File used to create grid node and boundary distances.	263
4	dual_plot	File used to plot permeability and streamlines together. Points are only plotted on the radial boundary intersections.	264
5	dual_plot_full	File used to plot permeability and streamlines together. All boundary intersection points are plotted.	265
6	face_velocities	File used to determine the face velocities from the node pressure solution.	266
7	flow_rate	File used to calculate flow rate using the approximate area method.	268
8	flow_rate_new	File used to calculate flow rate using the accurate cross product area method.	272
9	input_data	File used to input all required model data.	276
10	laplace_coeff	File used to calculate the Laplace coefficients.	277
11	laplace_solver	File used to create the system matrix and solve the Laplace equation.	282

No.	Filename	Description	Page
12	MAIN_FILE	File used to run the case studies requiring code in proper order. This is the main file used for each run.	287
13	permeability	File used to establish the various permeability cases.	288
14	porosity	File used to create the porosity matrix.	296
15	pressure_node_plots	File used to generate the 3D node pressure plots.	297
16	streamline_plotter	File used to create a polar plot of the streamlines. Points are only plotted on the radial boundary intersections.	299
17	streamline_plotter_full	File used to create a polar plot of the streamlines. All boundary intersection points are plotted.	300
18	stremline_polluck	File used to make a single streamline. One entrance point must be given at the start of the file.	301
19	stremline_polluck_more_pts	File used to generate all streamlines throughout the model.	307
20	upscaled_mobility	File used for upscaling procedures.	315
21	vel_face_plot_R	File used to plot radial face velocities.	316
22	vel_face_plot_T	File used to plot tangential face velocities.	317
23	vel_face_T_fix	File used to remove very small numbers which may cause calculation errors.	318
24	vel_field_coeff	File used to determine value of velocity field coefficients.	319

Code Name: casing_perm_fix

Code Number: 01

```
% Casing fix file
i = 1;
j = 1;
while j <= J
    if K_isotropic_blocks(1,j) == casing_perm
        node_pressures_ij_pa(1,j) = node_pressures_ij_pa(2,j);
    end
    j=j+1;
end
```

Code Name: darcy_flow_check

Code Number: 02

```
% file for checking flow rate calc vs values excepted from radial
% Darcy's Law
skin_nomonial = ((bulk_perm/perm_damage) - 1)*log(rad_damage/r_w);
rad_flow_check_noskin = 2*pi()*bulk_perm*h*(p_e-
p_wf)/mu_o/(log(r_e/r_w));
rad_flow_check_noskin_bbld = rad_flow_check_noskin*60*60*24*6.28994;
skin_calc = 2*pi()*bulk_perm*h*(p_e-p_wf)/(flow_rate_total_new*mu_o) -
log(r_e/r_w);
skin_calc_perf = skin_calc - skin_nomonial;
```

Code Name: distances

Code Number: 03

```
% Initial grid Discretization
%----------DATA LOADING BEGIN----------
input_data;                             % Input data file
%-----------DATA LOADING ENDS----------
% Calculating radius such that pressure drop is constant for
% a homogenous and isotropic cylinder
% Calculating such that r_0 is r_w, r_n+1/2 is r_e,
% and solving for nodes in between
R_nodes = ones(1,N);
i = 0;
while i < N
    R_nodes(1,i+1) = r_w*(r_e/r_w)^((i+1)/(N+0.5));    %
r_w*(r_e/r_w)^((i+1)/(N+0.5));
    i = i+1;
end
% The above R matrix determines only node locations from 1=1,..,N
% Creating boundary distances
% There is one more boundary than there are nodes (N+1)
% First boundary is rw or r1-1/2
% Second boundary is r1+1/2
% Therefore value (i) in boundary matrix will be the boundary before
% node in boundary matrix at position (i)
% Last boundary is re or rN+1/2
R_boundary = ones(1,N+1);
R_boundary(1,1) = r_w;
R_boundary(1,N+1) = r_e;
i = 2;
while i <= N
    R_boundary(1,i) = (R_nodes(1,i)-R_nodes(1,i-
1))/log((R_nodes(1,i))/(R_nodes(1,i-1)));
    i = i+1;
end
%creating thetas for nodes
theta_nodes = linspace(2*pi/(2*J),2*pi-2*pi/(2*J),J);
delta_theta = 360/J * pi/180;
```

Code Name: dual_plot

Code Number: 04

```
% Perm and streamline plot
%----------DATA LOADING BEGIN----------
%streamline_plotter;                    % Streamline in polar coord
%----------DATA LOADING ENDS----------
dimens_stream_matrix = size(stream_line_x_short);
stream_line_x_cart =
ones(dimens_stream_matrix(1,1),dimens_stream_matrix(1,2));
i = 1;
j = 1;
while i <= dimens_stream_matrix (1,1)
    while j <= dimens_stream_matrix (1,2)
        stream_line_x_cart(i,j) =
stream_line_x_short(i,j)*cos(stream_line_x_short(i,j+1));
        stream_line_x_cart(i,j+1) =
stream_line_x_short(i,j)*sin(stream_line_x_short(i,j+1));
        j = j+2;
    end
    i = i+1;
    j = 1;
end
plot_count = 1;
while plot_count <= total_pts
    plot(stream_line_x_cart(:,plot_count*2-
1),stream_line_x_cart(:,plot_count*2));
    hold all
    plot_count = plot_count + 1;
end
hold all
polarcont(R_nodes(1,:),theta_nodes(1,:),K_isotropic_blocks(:,:));
%surf(nodes_x_full,nodes_y_full,log(abs(K_isotropic_blocks_full)));
```

Code Name: dual_plot_full

Code Number: 05

```
% Perm and streamline plot full
%----------DATA LOADING BEGIN----------
%streamline_plotter;                    % Streamline in polar cord
%----------DATA LOADING ENDS----------
dimens_stream_matrix = size(stream_line_x);
stream_line_x_cart =
ones(dimens_stream_matrix(1,1),dimens_stream_matrix(1,2));
i = 1;
j = 1;
while i <= dimens_stream_matrix (1,1)
    while j <= dimens_stream_matrix (1,2)
        stream_line_x_cart(i,j) =
stream_line_x(i,j)*cos(stream_line_x(i,j+1));
        stream_line_x_cart(i,j+1) =
stream_line_x(i,j)*sin(stream_line_x(i,j+1));
        j = j+2;
    end
    i = i+1;
    j = 1;
end
plot_count = 1;
while plot_count <= total_pts
    plot(stream_line_x_cart(:,plot_count*2-
1),stream_line_x_cart(:,plot_count*2));
    hold all
    plot_count = plot_count + 1;
end
hold all
%polarcont(R_nodes(1,:),theta_nodes(1,:),K_isotropic_blocks(:,:));
surf(nodes_x_full,nodes_y_full,log(abs(K_isotropic_blocks_full)));
```

Code Name: face_velocities

Code Number: 06

```
% determining face velocities
%----------DATA LOADING BEGIN----------
laplace_solver;                         % Node pressures
pressure_node_plots;                    % Pressure plots
%-----------DATA LOADING ENDS----------
% changing node pressures back to Pa
node_pressures_ij_pa = node_pressures_ij * 10^5;
%----------DATA LOADING BEGIN----------
%casing_perm_fix;                       % Casing pressure fix
%-----------DATA LOADING ENDS----------
% there should be one more face velocity in the radial direction than
% the number of nodes since there is one more face than the number of
% nodes
vel_face_R = zeros(N+1,J);
% In the tangential direction there should be equal face velocities as
% there are nodes
vel_face_T = zeros(N,J);
% determining proper values to fill in for radial velocities
i = 1;
j = 1;
while i <= N
    while j <= J
        if i == 1
            vel_face_R(1,j)= mobility_upscale_R(1,j) *
(node_pressures_ij_pa(1,j)-p_wf)/(R_nodes(1,i)-r_w);
        end
        if i > 1 && i <= N
            vel_face_R(i,j)= mobility_upscale_R(i,j) *
(node_pressures_ij_pa(i,j)-node_pressures_ij_pa(i-1,j))/(R_nodes(1,i)-
R_nodes(1,i-1));
        end
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% for external boundary
while i <= N+1
    while j <= J
        vel_face_R(i,j)= mobility_upscale_R(i,j) * (p_e-
node_pressures_ij_pa(i-1,j))/(r_e-R_nodes(1,i-1));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
%-------------------------------------------------------------------------
% for tangential velocities
i = 1;
j = 1;
```

```
while i <= N
    while j <= J
        if j == 1
            vel_face_T(i,j)= mobility_upscale_T(i,j) *
(node_pressures_ij_pa(i,j)-
node_pressures_ij_pa(i,J))/(R_nodes(1,i)*delta_theta);
        end
        if j > 1
             vel_face_T(i,j)= mobility_upscale_T(i,j) *
(node_pressures_ij_pa(i,j)-node_pressures_ij_pa(i,j-
1))/(R_nodes(1,i)*delta_theta);
        end
        j = j+1;
    end
    i = i+1;
    j = 1;
end
%-------------------------------------------------------------------------
```

Code Name: flow_rate

Code Number: 07

```
% trying to set up first flow rate calc
%----------DATA LOADING BEGIN----------
%streamline_pollock_more_pts;                    % streamline runner
%----------DATA LOADING ENDS----------
size_stream = size(flow_rate_data_x);
time_travel_ring = zeros(N,total_pts);
count = 1;
i = N;
move = 1;
index_shift = 1;
placer = 1;
while index_shift <= 3*total_pts
    while count < size_stream(1,1) && i > 0
        while flow_rate_data_x(count,index_shift+1) == i  && count <
size_stream(1,1)
            time_travel_ring(move,placer) =
time_travel_ring(move,placer) + flow_rate_data_x(count,index_shift);
            count = count + 1;
        end
        i = i-1;
        move = move+1;
    end
    index_shift = index_shift+3;
    placer = placer+1;
    count = 1;
    i = N;
    move = 1;
end
% Shortening all streamlines such that we have a matrix with values
% only for radial ring intersections (no rows = no rings)
stream_line_x_short = zeros(N+1,total_pts*2);
i=N;
count = 1;
up_count = 1;
placer = 1;
while count <= 2*total_pts
    while up_count <= size(stream_line_x,1) &&
stream_line_x(up_count,count)>0          while
stream_line_x(up_count,count) ~= R_boundary(1,i+1)
            if stream_line_x(up_count,count) < r_w
                fprintf('error in a streamline ending point')
                stream_line_x_short(placer,count) =
stream_line_x(up_count,count);
                stream_line_x_short(placer,count+1) =
stream_line_x(up_count,count+1);
            end
            if stream_line_x(up_count,count) > r_w
                up_count = up_count+1;
            end
        end
```

```
        if stream_line_x(up_count,count) == R_boundary(1,i+1)
            stream_line_x_short(placer,count) =
stream_line_x(up_count,count);
            stream_line_x_short(placer,count+1) =
stream_line_x(up_count,count+1);
            up_count = up_count+1;
            i=i-1;
            placer = placer +1;
        end
    end
    count = count + 2;
    up_count = 1;
    placer = 1;
    i=N;
end
% creating streamline segment length matrix
stream_line_segment_lengths = zeros(N,total_pts);
count_a = 1;
count_b = 1;
ref_placer = 1;
while count_b*2 <= size(stream_line_x_short,2)
    while count_a < N+1
        stream_line_segment_lengths(count_a,count_b)=
sqrt(stream_line_x_short(count_a,ref_placer)^2+stream_line_x_short(coun
t_a+1,ref_placer)^2-
2*stream_line_x_short(count_a,ref_placer)*stream_line_x_short(count_a+1
,ref_placer)*cos(stream_line_x_short(count_a,ref_placer+1)-
stream_line_x_short(count_a+1,ref_placer+1)));
        count_a = count_a+1;
    end
    count_a = 1;
    count_b = count_b+1;
    ref_placer = ref_placer+2;
end
% creating matrix for distance between streamlines
stream_line_dist_btwn = zeros(N+1,total_pts);
count_a = 1;
count_b = 1;
ref_placer = 1;
while count_b*2 <= size(stream_line_x_short,2)
    while count_a <= N+1
        if ref_placer == 1
            stream_line_dist_btwn(count_a,count_b)=
stream_line_x_short(count_a,ref_placer)*sqrt(2*(1-
cos(stream_line_x_short(count_a,ref_placer+1)-
stream_line_x_short(count_a,2*total_pts))));
            count_a = count_a+1;
        end
        if ref_placer ~= 1
            stream_line_dist_btwn(count_a,count_b)=
stream_line_x_short(count_a,ref_placer)*sqrt(2*(1-
cos(stream_line_x_short(count_a,ref_placer+1)-
stream_line_x_short(count_a,ref_placer-1))));
            count_a = count_a+1;
        end
    end
    count_a = 1;
```

```
    count_b = count_b+1;
    ref_placer = ref_placer+2;
end
% Creating averages for distances in between
% first for distance between streamlines for area calculation
avg_sl_dist_btwn = zeros(N,total_pts);
i = 1;
j = 1;
while i <= N
    while j <= total_pts
        avg_sl_dist_btwn(i,j) = (stream_line_dist_btwn(i,j) +
stream_line_dist_btwn(i+1,j))/2;
        j=j+1;
    end
    i=i+1;
    j=1;
end
% averaging segment streamline lengths
avg_sl_segment_lengths = zeros(N,total_pts);
i = 1;
j = 1;
while i <= N
    while j < total_pts
        avg_sl_segment_lengths(i,j) = (stream_line_segment_lengths(i,j)
+ stream_line_segment_lengths(i,j+1))/2;
        j=j+1;
    end
    while j == total_pts
        avg_sl_segment_lengths(i,j) = (stream_line_segment_lengths(i,j)
+ stream_line_segment_lengths(i,1))/2;
        j=j+1;
    end
    i=i+1;
    j=1;
end
% creating total time travel for each streamline
total_stream_travel_time = zeros(1,total_pts);
i = 1;
j = 1;
while i <= N
    while j <= total_pts
        total_stream_travel_time(1,j) =
total_stream_travel_time(1,j)+time_travel_ring(i,j);
        j=j+1;
    end
    i=i+1;
    j=1;
end
% creating average total time travel for adjacent streamlines
avg_total_sl_travel_time = zeros(1,total_pts);
i = 1;
j = 1;
while i <= N
    while j < total_pts
        avg_total_sl_travel_time(1,j) =
(total_stream_travel_time(1,j)+total_stream_travel_time(1,j+1))/2;
        j=j+1;
```

```
    end
    while j == total_pts
        avg_total_sl_travel_time(1,j) =
(total_stream_travel_time(1,j)+total_stream_travel_time(1,1))/2;
        j=j+1;
    end
    i=i+1;
    j=1;
end
% determining flow rate between each streamline
flow_rate_streams = zeros(1,total_pts);
i = 1;
j = 1;
while i <= N
    while j <= total_pts
        flow_rate_streams(1,j) =
flow_rate_streams(1,j)+(poro_const/avg_total_sl_travel_time(1,j))*avg_s
l_dist_btwn(i,j)*h*avg_sl_segment_lengths(i,j);
        j=j+1;
    end
    i=i+1;
    j=1;
end
% determining total flow rate
flow_rate_total = 0;
j = 1;
while j <= total_pts
    flow_rate_total = flow_rate_total + flow_rate_streams(1,j);
    j=j+1;
end
flow_rate_total_bblperd = flow_rate_total*60*60*24*6.28994;
```

Code Name: flow_rate_new

Code Number: 08

```
% new flow rate calculation file using cross products to determine the
% flow rate creating streamline splitting segment length matrix
stream_line_segment_length_split = zeros(N,total_pts);
count_a = 1;
count_b = 1;
ref_placer = 1;
while count_b*2 <= size(stream_line_x_short,2)
    if ref_placer == 1
        while count_a < N+1
        stream_line_segment_length_split(count_a,count_b)=
sqrt(stream_line_x_short(count_a,ref_placer)^2+stream_line_x_short(coun
t_a+1,2*total_pts-1)^2-
2*stream_line_x_short(count_a,ref_placer)*stream_line_x_short(count_a+1
,2*total_pts-1)*cos(stream_line_x_short(count_a,ref_placer+1)-
stream_line_x_short(count_a+1,2*total_pts)));
        count_a = count_a+1;
        end
    end
    if ref_placer ~= 1
        while count_a < N+1
        stream_line_segment_length_split(count_a,count_b)=
sqrt(stream_line_x_short(count_a,ref_placer)^2+stream_line_x_short(coun
t_a+1,ref_placer-2)^2-
2*stream_line_x_short(count_a,ref_placer)*stream_line_x_short(count_a+1
,ref_placer-2)*cos(stream_line_x_short(count_a,ref_placer+1)-
stream_line_x_short(count_a+1,ref_placer-1)));
        count_a = count_a+1;
        end
    end
    count_a = 1;
    count_b = count_b+1;
    ref_placer = ref_placer+2;
end
% calculating first inside angle
stream_line_angle_1 = zeros(N,total_pts); %exterior
stream_line_angle_2 = zeros(N,total_pts); %interior
count_1 = 1;
count_2 = 1;
while count_1 <= total_pts
   while count_2 <= N
       if stream_line_dist_btwn(count_2+1,count_1) == 0 % special case
for streamlines on top of each other at perfs
           stream_line_angle_1(count_2,count_1) = pi()/2;
       end
       if stream_line_dist_btwn(count_2+1,count_1) ~= 0
           stream_line_angle_1(count_2,count_1) =
acos((stream_line_segment_lengths(count_2,count_1)^2+stream_line_dist_b
twn(count_2+1,count_1)^2-
stream_line_segment_length_split(count_2,count_1)^2)/(2*stream_line_seg
```

```
ment_lengths(count_2,count_1)*stream_line_dist_btwn(count_2+1,count_1))
);
        end
        count_2 = count_2 + 1;
    end
    count_2 = 1;
    count_1 = count_1 + 1;
end
% calculating 2nd outside angle
count_1 = 1;
count_2 = 1;
while count_1 <= total_pts
    if count_1 == 1
        while count_2 <= N
            if stream_line_dist_btwn(count_2,count_1) == 0 % special
case for streamlines on top of each other at perfs
                stream_line_angle_2(count_2,count_1) = pi()/2;
            end
            if stream_line_dist_btwn(count_2,count_1) ~= 0
                stream_line_angle_2(count_2,count_1) =
acos((stream_line_segment_lengths(count_2,total_pts)^2+stream_line_dist
_btwn(count_2,count_1)^2-
stream_line_segment_length_split(count_2,count_1)^2)/(2*stream_line_seg
ment_lengths(count_2,total_pts)*stream_line_dist_btwn(count_2,count_1))
);
            end
            count_2 = count_2 + 1;
        end
    end
    if count_1 ~= 1
        while count_2 <= N
            if stream_line_dist_btwn(count_2,count_1) == 0 % special
case for streamlines on top of each other at perfs
                stream_line_angle_2(count_2,count_1) = pi()/2;
            end
            if stream_line_dist_btwn(count_2,count_1) ~= 0
                stream_line_angle_2(count_2,count_1) =
acos((stream_line_segment_lengths(count_2,count_1-
1)^2+stream_line_dist_btwn(count_2,count_1)^2-
stream_line_segment_length_split(count_2,count_1)^2)/(2*stream_line_seg
ment_lengths(count_2,count_1-
1)*stream_line_dist_btwn(count_2,count_1)));
            end
           count_2 = count_2 + 1;
        end
    end
    count_2 = 1;
    count_1 = count_1 + 1;
end
% calculating area 1
stream_area_new_1 = zeros(N,total_pts); %exterior
stream_area_new_2 = zeros(N,total_pts); %interior
count_1 = 1;
count_2 = 1;
while count_1 <= total_pts
    while count_2 <= N
```

```
        if count_1 == 1
            stream_area_new_1(count_2,count_1) =
stream_line_segment_lengths(count_2,count_1)*stream_line_dist_btwn(coun
t_2+1,total_pts)*sin(stream_line_angle_1(count_2,count_1));
        end
        if count_1 ~= 1
            stream_area_new_1(count_2,count_1) =
stream_line_segment_lengths(count_2,count_1)*stream_line_dist_btwn(coun
t_2+1,count_1-1)*sin(stream_line_angle_1(count_2,count_1));
        end
        count_2 = count_2 + 1;
    end
    count_2 = 1;
    count_1 = count_1 + 1;
end
% calculating area 2
count_1 = 1;
count_2 = 1;
while count_1 <= total_pts
    while count_2 <= N
        if count_1 == 1
            stream_area_new_2(count_2,count_1) =
stream_line_segment_lengths(count_2,total_pts)*stream_line_dist_btwn(co
unt_2,total_pts)*sin(stream_line_angle_2(count_2,count_1));
        end
        if count_1 ~= 1
            stream_area_new_2(count_2,count_1) =
stream_line_segment_lengths(count_2,count_1-
1)*stream_line_dist_btwn(count_2,count_1-
1)*sin(stream_line_angle_2(count_2,count_1));
        end
        count_2 = count_2 + 1;
    end
    count_2 = 1;
    count_1 = count_1 + 1;
end
% calculating average area
stream_area_new_avg = zeros(N,total_pts);
count_1 = 1;
count_2 = 1;
while count_1 <= total_pts
    while count_2 <= N
        stream_area_new_avg(count_2,count_1) =
(stream_area_new_2(count_2,count_1)+stream_area_new_1(count_2,count_1))
/2;
        count_2 = count_2 + 1;
    end
    count_2 = 1;
    count_1 = count_1 + 1;
end
% determining flow rate between each streamline
flow_rate_streams_new = zeros(1,total_pts);
i = 1;
j = 1;
while i <= N
    while j <= total_pts
```

```
        flow_rate_streams_new(1,j) =
flow_rate_streams_new(1,j)+(poro_const/avg_total_sl_travel_time(1,j))*s
tream_area_new_avg(i,j)*h;
        j=j+1;
    end
    i=i+1;
    j=1;
end
% determining total flow rate
flow_rate_total_new = 0;
j = 1;
while j <= total_pts
    flow_rate_total_new = flow_rate_total_new +
flow_rate_streams_new(1,j);
    j=j+1;
end
flow_rate_total_new_bblperd = flow_rate_total_new*60*60*24*6.28994;
```

Code Name: input_data

Code Number: 09

```
% Input data
r_w = 0.05;                 % Wellbore radius
r_e = 50;                   % External (reservoir) radius
Blocks_rad = 300;           % Number of grid blocks in radial direction
Blocks_theta = 100;         % Number of grid blocks in theta direction
mu_o = 0.8e-3;              % oil viscosity (constant)
N = Blocks_rad;
J = Blocks_theta;
J_per_block = 1;            % number of start points per block
J_en = J_per_block;         % Number of entrance points per block
p_e = 300;                  % External reservoir pressure (bar)
p_wf = 280;                 % Flowing bottom-hole pressure (bar)
p_e = p_e * 10^5;           % Converting from bar to Pa
p_wf = p_wf * 10^5;         % Converting from bar to Pa
perm_case = 10;             % perm case number
bulk_perm = 1e-13;
poro_const = 0.25;
% perf case 9 data
rad_damage = 0.30;
perm_damage = 0.15e-13;
perf_thick = 0.04;
perf_length = 0.60;
perf_perm = 1000e-15;
perf_no = 3;
casing_perm = 1e-20;
% perf case 10 data
free_space_perm = 1e-10;
filter_perm = 165.8e-18; %x-18 is micro Darcy
FC_thickness = 9.0e-3;   %x-3 is mm
FC_coverage_angle = 270;
h = 5;                      % layer thickness (m)
toro = 0.40;                % tortuosity
grain_dia = 0.00025;        % average grain diameter (m)
kozo_const = 2e-6;          % kozeny constant
```

Code Name: laplace_coeff

Code Number: 10

```
% Creating file for Laplace Coefficients
% Note i,j corresponds with row, column
%----------DATA LOADING BEGIN----------
input_data;                             % Input data file
permeability;                           % Node permeability matrix
distances;                              % Node spacing
upscaled_mobility;                      % Grid mobility's
%-----------DATA LOADING ENDS----------
% First calculating coeff's for i=1,j=1,...,M;
% a coeff
a_coeff=ones(N,J);
i = 1;
j = 1;
while i == 1
    while j <= 1
        a_coeff(i,j) = -
((R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundar
y(1,i+1)-r_w)*(R_nodes(1,i+1)-R_nodes(1,i))) +
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-r_w)*(R_nodes(1,i)-r_w))+(mobility_upscale_T(i,j) +
mobility_upscale_T(i,J))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_
theta)));
        j = j+1;
    end
    while j <= J
        a_coeff(i,j) = -
((R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundar
y(1,i+1)-r_w)*(R_nodes(1,i+1)-R_nodes(1,i))) +
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-r_w)*(R_nodes(1,i)-r_w))+(mobility_upscale_T(i,j) +
mobility_upscale_T(i,j-
1))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_theta)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% b coeff
b_coeff=ones(N,J);
i = 1;
j = 1;
while i == 1
    while j <= J
        b_coeff(i,j) =
(mobility_upscale_T(i,j))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta
_theta));
        j = j+1;
    end
    i = i+1;
    j = 1;
```

```
end
% c coeff
c_coeff=ones(N,J);
i = 1;
j = 1;
while i == 1
    while j <= 1
        c_coeff(i,j) =
(mobility_upscale_T(i,J))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta
_theta));
        j = j+1;
    end
    while j <= J
        c_coeff(i,j) = (mobility_upscale_T(i,j-
1))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_theta));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% d coeff
d_coeff=ones(N,J);
i = 1;
j = 1;
while i == 1
    while j <= J
        d_coeff(i,j) =
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-r_w)*(R_nodes(1,i)-r_w));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% e coeff
e_coeff=ones(N,J);
i = 1;
j = 1;
while i == 1
    while j <= J
        e_coeff(i,j) =
(R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundary
(1,i+1)-r_w)*(R_nodes(1,i+1)-R_nodes(1,i)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% Now calculating coeff's for i=2,...,N-1; j=1,...,M;
% a coeff
i = 2;
j = 1;
while i < N
    while j <= 1
        a_coeff(i,j) = -
((R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundar
y(1,i+1)-R_boundary(1,i))*(R_nodes(1,i+1)-R_nodes(1,i))) +
```

```
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-R_boundary(1,i))*(R_nodes(1,i)-R_nodes(1,i-
1)))+(mobility_upscale_T(i,j) +
mobility_upscale_T(i,J))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_
theta)));
        j = j+1;
    end
    while j <= J
        a_coeff(i,j) = -
((R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundar
y(1,i+1)-R_boundary(1,i))*(R_nodes(1,i+1)-R_nodes(1,i))) +
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-R_boundary(1,i))*(R_nodes(1,i)-R_nodes(1,i-
1)))+(mobility_upscale_T(i,j) + mobility_upscale_T(i,j-
1))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_theta)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% b coeff
i = 2;
j = 1;
while i < N
    while j <= J
        b_coeff(i,j) =
(mobility_upscale_T(i,j))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta
_theta));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% c coeff
i = 2;
j = 1;
while i < N
    while j <= 1
        c_coeff(i,j) =
(mobility_upscale_T(i,J))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta
_theta));
        j = j+1;
    end
    while j <= J
        c_coeff(i,j) = (mobility_upscale_T(i,j-
1))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_theta));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% d coeff
i = 2;
j = 1;
while i < N
    while j <= J
```

```
        d_coeff(i,j) =
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-R_boundary(1,i))*(R_nodes(1,i)-R_nodes(1,i-1)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% e coeff
i = 2;
j = 1;
while i < N
    while j <= J
        e_coeff(i,j) =
(R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundary
(1,i+1)-R_boundary(1,i))*(R_nodes(1,i+1)-R_nodes(1,i)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% Now calculating coeff's for i=N; j=1,...,M;
% a coeff
i = N;
j = 1;
while i == N
    while j <= 1
        a_coeff(i,j) = -
((R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundar
y(1,i+1)-R_boundary(1,i))*(r_e-R_nodes(1,i))) +
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-R_boundary(1,i))*(R_nodes(1,i)-R_nodes(1,i-
1)))+(mobility_upscale_T(i,j) +
mobility_upscale_T(i,J))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_
theta)));
        j = j+1;
    end
    while j <= J
        a_coeff(i,j) = -
((R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundar
y(1,i+1)-R_boundary(1,i))*(r_e-R_nodes(1,i))) +
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-R_boundary(1,i))*(R_nodes(1,i)-R_nodes(1,i-
1)))+(mobility_upscale_T(i,j) + mobility_upscale_T(i,j-
1))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_theta)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% b coeff
i = N;
j = 1;
while i == N
    while j <= J
```

```
        b_coeff(i,j) =
(mobility_upscale_T(i,j))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta
_theta));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% c coeff
i = N;
j = 1;
while i == N
    while j <= 1
        c_coeff(i,j) =
(mobility_upscale_T(i,J))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta
_theta));
        j = j+1;
    end
    while j <= J
        c_coeff(i,j) = (mobility_upscale_T(i,j-
1))/((R_nodes(1,i)*R_nodes(1,i)*delta_theta*delta_theta));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% d coeff
i = N;
j = 1;
while i == N
    while j <= J
        d_coeff(i,j) =
(R_boundary(1,i)*mobility_upscale_R(i,j))/(R_nodes(1,i)*(R_boundary(1,i
+1)-R_boundary(1,i))*(R_nodes(1,i)-R_nodes(1,i-1)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% e coeff
i = N;
j = 1;
while i == N
    while j <= J
        e_coeff(i,j) =
(R_boundary(1,i+1)*mobility_upscale_R(i+1,j))/(R_nodes(1,i)*(R_boundary
(1,i+1)-R_boundary(1,i))*(r_e-R_nodes(1,i)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
```

Code Name: laplace_solver

Code Number: 11

```
% File to solve Laplace equation
% Note i,j corresponds with row, column
%----------DATA LOADING BEGIN----------
input_data;                              % Input data file
permeability;                            % Node permeability matrix
distances;                               % Node spacing
upscaled_mobility;                       % Grid mobility's
laplace_coeff;                           % Laplace coefficients
%----------DATA LOADING ENDS----------
%creating system matrix containing coefficients
system_matrix = sparse(N*J,N*J);
count = 1;
% turning (i,j) coeff matrixes into column matrixes
% a_coeff change
a_coeff_column = ones(N*J,1);
i = 1;
j = 1;
 while i <= N;
     while j <= J;
         a_coeff_column(count,1) = a_coeff(i,j);
         j= j+1;
         if i == 1
             count = j;
         else
             count = (i-1)*J+j;
         end
     end
     i = i+1;
     j = 1;
 end
% b_coeff change
b_coeff_column = ones(N*J,1);
i = 1;
j = 1;
count = 1;
 while i <= N;
     while j <= J;
         b_coeff_column(count,1) = b_coeff(i,j);
         j= j+1;
         if i == 1
             count = j;
         else
             count = (i-1)*J+j;
         end
     end
     i = i+1;
     j = 1;
 end
% c_coeff change
c_coeff_column = ones(N*J,1);
```

```
i = 1;
j = 1;
count = 1;
 while i <= N;
     while j <= J;
         c_coeff_column(count,1) = c_coeff(i,j);
         j= j+1;
         if i == 1
             count = j;
         else
             count = (i-1)*J+j;
         end
     end
     i = i+1;
     j = 1;
 end
% d_coeff change
d_coeff_column = ones(N*J,1);
i = 1;
j = 1;
count = 1;
 while i <= N;
     while j <= J;
         d_coeff_column(count,1) = d_coeff(i,j);
         j= j+1;
         if i == 1
             count = j;
         else
             count = (i-1)*J+j;
         end
     end
     i = i+1;
     j = 1;
 end
% e_coeff change
e_coeff_column = ones(N*J,1);
i = 1;
j = 1;
count = 1;
 while i <= N;
     while j <= J;
         e_coeff_column(count,1) = e_coeff(i,j);
         j= j+1;
         if i == 1
             count = j;
         else
             count = (i-1)*J+j;
         end
     end
     i = i+1;
     j = 1;
 end
 % filling in system matrix containing coefficients
 % first filling in a_coeff
i = 1;
j = 1;
count = 1;
```

```
while count <= N*J
    system_matrix(count,count) = a_coeff_column(count,1);
    count = count + 1;
end
% filling in b_coeff
i = 1;
j = 1;
count = 1;
count_b = 1;
while count <= N*J
    while i <= N;
        while j <= J
            if i ==1
                count = j;
            else
                count = (i-1)*J+j;
            end
            if count == i*J;
                count_b = count + 1 - J;
            else
                count_b = count + 1;
            end
            system_matrix(count,count_b) = b_coeff_column(count,1);
            j = j+1;
        end
        i = i+1;
        j = 1;
    end
    count = count + 1;
end
% filling in c_coeff
i = 1;
j = 1;
count = 1;
count_c = 1;
while count <= N*J
    while i <= N;
        while j <= J
            if i == 1
                count = j;
            else
                count = (i-1)*J+j;
            end
            if count == (((i-1)*J) + 1)
                count_c = count - 1 + J;
            else
                count_c = count - 1;
            end
            system_matrix(count,count_c) = c_coeff_column(count,1);
            j = j+1;
        end
        i = i+1;
        j = 1;
    end
    count = count + 1;
end
% filling in d_coeff
```

```
i = 2;  %Starting at i = 2 to skip i = 1
j = 1;
count = 1;
count_d = 1;
while count <= N*J
    while i <= N;
        while j <= J
            count = (i-1)*J+j;
            count_d = count - J;
             system_matrix(count,count_d) = d_coeff_column(count,1);
            j = j+1;
        end
        i = i+1;
        j = 1;
    end
    count = count + 1;
end
% filling in e_coeff
i = 1;
j = 1;
count = 1;
count_e = 1;
while count <= N*J
    while i < N;    % i < N instead of i<=N avoid last i values
        while j <= J
            count = (i-1)*J+j;
            count_e = count + J;
            system_matrix(count,count_e) = e_coeff_column(count,1);
            j = j+1;
        end
        i = i+1;
        j = 1;
    end
    count = count + 1;
end
% creating known value matrix from boundary conditions
boundary_matrix = zeros(N*J,1);
i = 1;
j = 1;
count = 1;
count_bound = 1;
while count <= N*J
    while i <= N;    % i < N instead of i<=N avoid last i values
        while j <= J
            count = (i-1)*J+j;
            if i==1
                boundary_matrix(count,1) = -1 *
d_coeff_column(count,1)*p_wf;
            end
            if i==N
                boundary_matrix(count,1) = -1 *
e_coeff_column(count,1)*p_e;
            end
            j = j+1;
        end
        i = i+1;
        j = 1;
```

```
    end
    count = count + 1;
end
% creating unknown node pressure matrix
node_pressures = system_matrix\boundary_matrix;
% converting pressures from Pa to bar
node_pressures = node_pressures / 10^5;
```

Code Name: MAIN_FILE

Code Number: 12

```
% Main running files
% This file run the required files for proper analysis of a cases
% Select the proper permeability case and input the desired data in the
% input file before running this file
%----------DATA LOADING BEGIN----------
stremline_polluck_more_pts;          % runs the Laplace and streamline
generators
flow_rate;                           % determines approximate flow rate
flowrate_new;                        % determines actual flow rate
darcy_flow_check;                    % determines the base case flow and
resulant case skin
%----------DATA LOADING ENDS----------
flow_rate_total_new_bblperd
skin_calc
```

Code Name: permeability

Code Number: 13

```
% Creating file for permeability
% Note i,j corresponds with row, column
%----------DATA LOADING BEGIN----------
input_data;                              % Input data file
distances;                               % input node and boundary distances
%----------DATA LOADING ENDS----------
% This project will use a mapped isotropic medium to be considered
% later
if perm_case == 1
    %-----------------------------------------
    %------Case 1 = perm decrease with r------
    %-----------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
    Perm_mult_fact = ones(N,1);
    i=1;
    while i <= N;
       Perm_mult_fact(i,1) = (i/N)* Perm_mult_fact(i,1);
       K_isotropic_blocks(i,:) = K_isotropic_blocks(i,:)*
Perm_mult_fact(i,1);
       i=i+1;
    end
end
if perm_case == 2
    %-----------------------------------------
    %----------Case 2 = Homogenous------------
    %-----------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
end
if perm_case == 3
    %-----------------------------------------
    %-----Case 3 = Sector of high perm--------
    %-----------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
    i=1;
    j=floor(3*J/8);
    while i <= N
        while j < J/2 && j >= floor(3*J/8)
            if i <= N-1 && i >= 1
                K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*1.5;
            end
            j=j+1;
        end
       i=i+1;
       j=floor(3*J/8);
```

```
    end
end
if perm_case == 4
    %-----------------------------------
    %---Case 4 = Section of low Perm (ring)--
    %-----------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
    i=1;
    j=1;
    while i <= N
        while j <= J
            if i <= 3*N/4 && i >= 2*N/3
                K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.1;
            end
            j=j+1;
        end
       i=i+1;
       j=1;
    end
end
if perm_case == 5
%-----------------------------------
%------Case 5 = block of low perm--------
%-----------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
    i=1;
    j=floor(3*J/8);
    while i <= N
       while j < J/2 && j >= floor(3*J/8)
            if i <= 2.75*N/3 + 2 && i >= 2.5*N/3
               K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.25;
            end
            j=j+1;
        end
       i=i+1;
       j=floor(3*J/8);
    end
end
if perm_case == 6
    %-----------------------------------
    %-----Case 6 = Several sector of low perm--------
    %-----------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
    i=1;
    j=floor(3*J/8);
    while i <= N
        while j < J/2 && j >= floor(3*J/8)
            if i <= N-1 && i >= 1
               K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*100;
            end
            j=j+1;
```

```
        end
        i=i+1;
        j=floor(3*J/8);
    end
    i=1;
    j=floor(5*J/8);
    while i <= N
        while j < 6*J/8 && j >= floor(5*J/8)
            if i <= N-1 && i >= 1
                K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*100;
            end
            j=j+1;
        end
        i=i+1;
        j=floor(5*J/8);
    end
end
if perm_case == 7
%-----------------------------------------
%------Case 7 = block of low perm---------
%-----------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
    i=1;
    j=floor(5*J/8);
    while i <= N
        while j < 6*J/8 && j >= floor(5*J/8)
            if i <= 2.5*N/3 && i >= N/2
                K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.05;
            end
            j=j+1;
        end
        i=i+1;
        j=floor(5*J/8);
    end
    i=1;
    j=floor(5*J/8);
    while i <= N
        while j < J/2 && j >= floor(3*J/8)
            if i <= floor(2.85*N/3) && i >= floor(1.5*N/2)
                K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.05;
            end
            j=j+1;
        end
        i=i+1;
        j=floor(3*J/8);
    end
end
if perm_case == 8
    %-----------------------------------------
    %-----Case 8 = Sector of low perm near start--------
    %-----------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
```

```
    i=1;
    j=floor(1*J/8);
    while i <= N
        while j < J/4 && j >= floor(1*J/8)
            if i <= N-1 && i >= 1
                K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*100;
            end
            j=j+1;
        end
       i=i+1;
       j=floor(1*J/8);
    end
end
if perm_case == 9
    %---------------------------------------
    %-----Case 9 = Damage, casing and perfs--------
    %---------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2
    % determining damage zone
    i_damage = ceil((N+0.5)*log(rad_damage/r_w)/log(r_e/r_w) - 1);
    i = 1;
    j = 1;
    while i <= i_damage
        while j <= J
            K_isotropic_blocks(i,j) = perm_damage;
            j=j+1;
        end
        j=1;
        i=i+1;
    end
    % adding casing
    i = 1;
    j = 1;
        while j <= J
            K_isotropic_blocks(1,j) = casing_perm;
            j=j+1;
        end
    % determining perfs
    i_perf = ceil((N+0.5)*log(perf_length/r_w)/log(r_e/r_w) - 1);
    i = 1;
    j = 1;
    J_start_perf=ones(1,perf_no);
    pcount = 1;
    while pcount <= perf_no
        J_start_perf(1,pcount) = pcount*floor(J/perf_no)-
floor(J/(perf_no*2));
        pcount = pcount + 1;
    end
   while i <= i_perf
        J_perf_wide(1,i) =
floor(perf_thick/(delta_theta*R_nodes(1,i)));
       i = i+1;
   end
   i = 1;
   x = 1;
```

```
    j = J_start_perf(1,x);
    while x <= perf_no
        while i <= i_perf
            j=J_start_perf(1,x);
             while j < J_start_perf(1,x)+J_perf_wide(1,i) && j >=
J_start_perf(1,x)
                 K_isotropic_blocks(i,j) = perf_perm;
                 j=j+1;
             end
            i=i+1;
            j=J_start_perf(1,x);
        end
        i = 1;
        x = x+1;
    end
 % checking for perforation length error
 perf_check = size(J_perf_wide,2);
 perf_end_width = R_nodes(1,perf_check)*delta_theta;
 if perf_end_width > perf_thick
     fprintf('ERROR --> perf end is greater than perf width. Increase
the number of J blocks or per will not be required length')
 end
end
% creating porosity matrix based on perm matrix - using Kozeny
poro = ones(N,J)*poro_const; % setting as constant for now
%i = 1;
%j = 1;
%while i <= N
%     while j <= J
%         poro(i,j) =
K_isotropic_blocks(i,j)*poro/(kozo_const*grain_dia^2);
%         j = j+1;
%     end
%     i = i+1;
%     j = 1;
%end
if perm_case == 10
    %---------------------------------------
    %-----Case 10 = Partial filter cake--------
    %---------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2
    % REMEMBER WELBORE SIZE MUST BE REDUCED
    % Setting interior of face to perm of free spaces
    j = 1;
        while j <= J
            while i < 9
                K_isotropic_blocks(i,j) = free_space_perm;
                i=i+1;
            end
             i= 1;
            j = j+1;
        end
    % adding filter cake
    %determine # blocks to work in from node 8
    if FC_thickness > 0
```

```
        num_fc_node = 1;
        while R_boundary(1,9) - R_boundary(1,9-num_fc_node) <
FC_thickness
            num_fc_node = num_fc_node + 1;
        end
    end
    first_fc_node = 9 - num_fc_node;
    FC_j_start = 1;
    FC_j_end = round(FC_coverage_angle/360*J);
    i = first_fc_node;
    j = 1;
        while j <= FC_j_end && j >= FC_j_start
            while i < 9
                K_isotropic_blocks(i,j) = filter_perm;
                i=i+1;
            end
            i=first_fc_node;
            j=j+1;
        end
     j = 1;
end
if perm_case == 11
%---------------------------------------
%------Case 11 = small blocks of low perm--------
%---------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm; % making all
blocks = 1 Darcy (in m^2)
    i=1;
    j=floor(1*J/16);
    while i <= N
       while j < floor(2*J/16) && j >= floor(1*J/16)
             if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
               K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
            end
            j=j+1;
        end
       i=i+1;
       j=floor(1*J/16);
    end
     i=1;
    j=floor(3*J/16);
    while i <= N
       while j < floor(4*J/16) && j >= floor(3*J/16)
            if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
               K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
            end
            j=j+1;
        end
       i=i+1;
       j=floor(3*J/16);
    end
        i=1;
    j=floor(5*J/16);
    while i <= N
       while j < floor(6*J/16) && j >= floor(5*J/16)
            if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
```

```
            K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
         end
         j=j+1;
      end
      i=i+1;
      j=floor(5*J/16);
   end
      i=1;
   j=floor(7*J/16);
   while i <= N
      while j < floor(8*J/16) && j >= floor(7*J/16)
         if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
            K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
         end
         j=j+1;
      end
      i=i+1;
      j=floor(7*J/16);
   end
  i=1;
   j=floor(9*J/16);
   while i <= N
      while j < floor(10*J/16) && j >= floor(9*J/16)

         if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
            K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
         end
         j=j+1;
      end
      i=i+1;
      j=floor(9*J/16);
   end
    i=1;
   j=floor(11*J/16);
   while i <= N
      while j < floor(12*J/16) && j >= floor(11*J/16)

         if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
            K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
         end
         j=j+1;
      end
      i=i+1;
      j=floor(11*J/16);
   end
      i=1;
   j=floor(13*J/16);
   while i <= N
      while j < floor(14*J/16) && j >= floor(13*J/16)
          if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
            K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
         end
         j=j+1;
      end
      i=i+1;
      j=floor(13*J/16);
```

```
    end
        i=1;
    j=floor(15*J/16);
    while i <= N
       while j < floor(16*J/16) && j >= floor(15*J/16)
            if i <= 2.5*N/3 + 2 && i >= 2.3*N/3
               K_isotropic_blocks(i,j) = K_isotropic_blocks(i,j)*0.75;
            end
            j=j+1;
        end
       i=i+1;
       j=floor(15*J/16);
    end
end
if perm_case == 12
    %-------------------------------------
    %-----Case 12 = Damage zone--------
    %-------------------------------------
    K_isotropic_blocks = ones(N,J);
    K_isotropic_blocks = K_isotropic_blocks * bulk_perm;
    % making all blocks = 1 Darcy (in m^2
    % determining damage zone
    i_damage = ceil((N+0.5)*log(rad_damage/r_w)/log(r_e/r_w) - 1);
    i = 1;
    j = 1;
    while i <= i_damage
        while j <= J
            K_isotropic_blocks(i,j) = perm_damage;
            j=j+1;
        end
        j=1;
        i=i+1;
    end
end
```

Code Name: porosity

Code Number: 14

```
% creating porosity matrix
poro_blocks = ones(N,J);
poro_blocks = poro_blocks * 0.2; % Make more complex later
```

Code Name: pressure_node_plots

Code Number: 15

```
% creating pressure plotting file
% first method will be to convert back to cart coordinates
% will try later in polar coordinates
%----------DATA LOADING BEGIN----------
input_data;                          % Input data file
permeability;                        % Node permeability matrix
distances;                           % Node spacing
upscaled_mobility;                   % Grid mobility's
laplace_solver;                      % Node pressures
%-----------DATA LOADING ENDS----------
% determining x,y coordinates of nodes
nodes_theta = linspace(2*pi/(2*J),2*pi-2*pi/(2*J),J);
nodes_x = zeros(N,J);
nodes_y = zeros(N,J);
i = 1;
j = 1;
while i <= N;
   while j <= J;
     nodes_x(i,j) = R_nodes(1,i)*cos(nodes_theta(1,j));
     nodes_y(i,j) = R_nodes(1,i)*sin(nodes_theta(1,j));
     j = j+1;
   end
   i = i+1;
   j = 1;
end
% arranging node pressure matrix to be i,j format to correspond with
nodes
node_pressures_ij = zeros(N,J);
i = 1;
j = 1;
count_ij = 1;
while i <= N;
   while j <= J;
       count_ij = (i-1)*J+j;
       node_pressures_ij(i,j) = node_pressures(count_ij,1);
     j = j+1;
   end
   i = i+1;
   j = 1;
end
% closing gap
i = 1;
j = 1;
nodes_x_full = zeros(N,J+1);
nodes_y_full = zeros(N,J+1);
node_pressures_ij_full = zeros(N,J+1);
K_isotropic_blocks_full = zeros(N,J+1);
while i<=N
    while j<=J
        nodes_x_full(i,j) = nodes_x(i,j);
```

```
        nodes_y_full(i,j) = nodes_y(i,j);
        node_pressures_ij_full(i,j) = node_pressures_ij(i,j);
        K_isotropic_blocks_full(i,j) = K_isotropic_blocks(i,j);
        j=j+1;
    end
    j = 1;
    i = i+1;
end
i = 1;
while i<=N
    nodes_x_full(i,J+1) = nodes_x(i,1);
    nodes_y_full(i,J+1) = nodes_y(i,1);
    node_pressures_ij_full(i,J+1) = node_pressures_ij(i,1);
    K_isotropic_blocks_full(i,J+1) = K_isotropic_blocks(i,1);
    i = i+1;
end
% creating 3D x,y,z plot
surf(nodes_x_full,nodes_y_full,node_pressures_ij_full)
%surf(nodes_x_full,nodes_y_full,K_isotropic_blocks_full)
```

Code Name: streamline_plotter

Code Number: 16

```
% stream_line plotting file
plot_count = 1;
while plot_count <= total_pts
polar(stream_line_x_short(:,plot_count*2),stream_line_x_short(:,plot_co
unt*2-1))
    hold all
    plot_count = plot_count + 1;
end
```

Code Name: streamline_plotter_full

Code Number: 17

```
% stream_line plotting file
plot_count = 1;
while plot_count <= total_pts
    polar(stream_line_x(:,plot_count*2),stream_line_x(:,plot_count*2-
1))
    hold all
    plot_count = plot_count + 1;
end
```

Code Name: stremline_polluck

Code Number: 18

```
% Basic streamline logic
%----------DATA LOADING BEGIN----------
%%%%%%%%%%vel_field_coeff;                    % Velocity coefficients
%----------DATA LOADING ENDS----------
% setting entrance point
stream_line = [r_e, 1.839577032];%(1.2/4*J)*delta_theta +
delta_theta/2];
theta_enter = stream_line(1,2);
if theta_enter > 2*pi()
    theta_enter = theta_enter-2*pi();
end
rad_enter = stream_line(1,1);
% determining i and j locations
j_current = ceil(stream_line(1,2)/delta_theta);
if j_current == J+1
    j_current = 1;
end
i_current = N;
rad_exit = stream_line(1,1)*0.99; % just for starting purposes
count = 1;
while rad_exit > r_w && rad_exit <= r_e
        % determining full sector distance travel times
        % determining theta dirn initial exit points
    if vel_face_T(i_current,j_current) > 0
        if j_current < J
            if vel_face_T(i_current,j_current+1) > 0 % case 1 - exit cw
                theta_exit = theta_boundaries(1,j_current);
            end
            if vel_face_T(i_current,j_current+1) < 0 % case 2 - moves
outward
                theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) *
delta_theta/(vel_face_T(i_current,j_current+1)-
vel_face_T(i_current,j_current));
                if theta_enter < theta_star
                    theta_exit = theta_boundaries(1,j_current);
                else
                    theta_exit = theta_boundaries(1,j_current+1);
                end
            end
            if vel_face_T(i_current,j_current+1) == 0 % case 1*
                theta_exit = theta_boundaries(1,j_current);
            end
        end
        if j_current == J % special case for boundary condition
            if vel_face_T(i_current,1) > 0 % case 1 - exit cw
                theta_exit = theta_boundaries(1,j_current);
            end
            if vel_face_T(i_current,1) < 0 % case 2 - moves outward
```

```
                theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) * delta_theta/(vel_face_T(i_current,1)-
vel_face_T(i_current,j_current));
                if theta_enter < theta_star
                    theta_exit = theta_boundaries(1,j_current);
                else
                    theta_exit = theta_boundaries(1,1);
                end
            end
        end
    end
    if vel_face_T(i_current,j_current) < 0
        if j_current < J
            if vel_face_T(i_current,j_current+1) < 0 % case 3 - exit
ccw
                theta_exit = theta_boundaries(1,j_current+1);
            end
            if vel_face_T(i_current,j_current+1) > 0 % case 4 - moves
inward
                theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) *
delta_theta/(vel_face_T(i_current,j_current+1)-
vel_face_T(i_current,j_current));
                theta_exit = theta_star;
            end
        end
        if j_current == J % special case for boundary condition
            if vel_face_T(i_current,1) < 0 % case 3 - exit ccw
                theta_exit = theta_boundaries(1,1);
            end
            if vel_face_T(i_current,1) > 0 % case 4 - moves inward
                theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) * delta_theta/(vel_face_T(i_current,1)-
vel_face_T(i_current,j_current));
                theta_exit = theta_star;
            end
        end
    end
    if vel_face_T(i_current,j_current) == 0 % radial case
        if j_current < J
            if vel_face_T(i_current,j_current+1) < 0 % case 3 - exit
ccw
                theta_exit = theta_boundaries(1,j_current+1);
            end
            if vel_face_T(i_current,j_current+1) > 0 % case 4 - moves
inward
                theta_exit = theta_boundaries(1,j_current);
            end
        end
        if j_current == J % special case for boundary condition
            if vel_face_T(i_current,1) < 0 % case 3 - exit ccw
                theta_exit = theta_boundaries(1,1);
            end
            if vel_face_T(i_current,1) > 0 % case 4 - moves inward
                theta_exit = theta_boundaries(1,j_current);
            end
        end
```

```
    end
    if theta_exit == theta_enter % radial
        time_theta = 1e50; % any large number (will look from min)
    else
        time_theta =
abs(1/vel_coeff_theta_a(i_current,j_current)*(log(vel_coeff_theta_b(i_c
urrent,j_current)+vel_coeff_theta_a(i_current,j_current)*theta_exit)-
log(vel_coeff_theta_b(i_current,j_current)+vel_coeff_theta_a(i_current,
j_current)*theta_enter)));
    end
    % determining radial dirn travel time
    if abs(vel_coeff_rad_b(i_current,j_current)) > 1e-10
        time_rad =
1/(vel_coeff_rad_b(i_current,j_current)^2)*(vel_coeff_rad_b(i_current,j
_current)*(rad_enter-R_boundary(1,i_current))-
vel_coeff_rad_a(i_current,j_current)*(log(vel_coeff_rad_a(i_current,j_c
urrent)+vel_coeff_rad_b(i_current,j_current)*rad_enter) -
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*R_boundary(1,i_current))));
    end
    if abs(vel_coeff_rad_b(i_current,j_current)) < 1e-10 % no leakage -
radial
        time_rad =
1/(2*vel_coeff_rad_a(i_current,j_current))*(rad_enter^2-
R_boundary(1,i_current)^2);
    end
    % determining minimum travel time
    time_min = min(time_theta,time_rad);
    % determining particle exit location
    if time_min == time_rad
        if abs(vel_coeff_theta_a(i_current,j_current)) < 1e-15
            theta_exit = theta_enter;
        end
        if abs(vel_coeff_theta_a(i_current,j_current)) > 1e-15
            theta_exit =
((vel_coeff_theta_b(i_current,j_current)+vel_coeff_theta_a(i_current,j_
current)*theta_enter)/exp(time_min*vel_coeff_theta_a(i_current,j_curren
t))-
vel_coeff_theta_b(i_current,j_current))/vel_coeff_theta_a(i_current,j_c
urrent);
        end
        rad_exit = R_boundary(1,i_current);
        i_current = i_current - 1;
        theta_enter = theta_exit;
        rad_enter = rad_exit;
    end
    if time_min == time_theta
        rad_exit_guess_1 =  R_boundary(1,i_current);% added
        i_current_1 = i_current;% added
        j_current_1 = j_current;% added
        rad_exit_guess =  R_boundary(1,i_current);
        NR_num = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
rad_exit_guess)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(
i_current,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_
coeff_rad_b(i_current,j_current)*rad_enter)-
```

```
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*rad_exit_guess));
        NR_num_1 = NR_num;% added
        NR_deno = (1/vel_coeff_rad_b(i_current,j_current)) -
(vel_coeff_rad_a(i_current,j_current)/vel_coeff_rad_b(i_current,j_curre
nt))*(1/(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current
,j_current)*rad_exit_guess));
        NR_deno_1 = NR_deno;% added
        rad_exit = rad_exit_guess - (NR_num/NR_deno);
        while abs(rad_exit-rad_exit_guess)/abs(rad_exit) > 0.00001
            rad_exit_guess = rad_exit;
            NR_num = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
rad_exit_guess)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(
i_current,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_
coeff_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*rad_exit_guess));
            NR_deno = (1/vel_coeff_rad_b(i_current,j_current)) -
(vel_coeff_rad_a(i_current,j_current)/vel_coeff_rad_b(i_current,j_curre
nt))*(1/(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current
,j_current)*rad_exit_guess));
            rad_exit = rad_exit_guess - (NR_num/NR_deno);
        end
        if isreal(rad_exit) == 0
            % instead of NR, usebisection-----------------------------
            guess_upper = R_boundary(1,i_current+1);
            guess_lower = R_boundary(1,i_current);
            answer_upper = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_upper)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_c
urrent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coe
ff_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_upper));
            answer_lower = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_lower)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_c
urrent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coe
ff_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_lower));
            product_guesses = answer_lower*answer_upper;
            if product_guesses > 0
                fprintf('ERROR --> Bisection not bounding answer')
            end
            if product_guesses < 0
                guess_new = (guess_upper+guess_lower)/2;
                answer_new_guess = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_new)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_cur
rent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coeff
_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_new));
            end
            if answer_new_guess*answer_upper < 0
```

```
                new_bound = guess_upper;
            end
            if answer_new_guess*answer_upper > 0
                new_bound = guess_lower;
            end
            while new_bound-guess_new > 0.00001
                answer_1 = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
new_bound)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_cur
rent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coeff
_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*new_bound));
                answer_2 = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_new)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_cur
rent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coeff
_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_new));
                if answer_1*answer_2 < 0
                    new_bound = new_bound;
                end
                if answer_1*answer_2 > 0
                    new_bound = guess_new;
                end
                guess_new = (guess_new+new_bound)/2;
            end
            rad_exit = guess_new;
        end
% end bisection-----------------------------------------------------------
        if rad_exit < r_w
            rad_exit = r_w;
        end
        if theta_exit > theta_enter
            j_current = j_current + 1;
            if j_current == J+1
                j_current = 1;
            end
        end
        if theta_exit < theta_enter
            j_current = j_current - 1;
            if j_current == 0
                j_current = J;
            end
        end
        rad_enter = rad_exit;
        theta_enter = theta_exit;
    end
    count = count+1;
    stream_line(count,1) = rad_exit;
    stream_line(count,2) = theta_exit;
    stream_line(count,3) = time_rad;
    stream_line(count,4) = time_theta;
    stream_line(count,5) = time_min;
    % forcing casing hits to go to perf
```

```
    if rad_exit == R_boundary(1,2) &&
K_isotropic_blocks(i_current,j_current) == casing_perm
        if vel_face_T(i_current+1,j_current) < 0
            block_count = 0;
            j_str = j_current;
            vel_start = vel_face_T(i_current,j_current);
            while K_isotropic_blocks(i_current,j_current) ~= perf_perm
                block_count = block_count+1;
                j_current = j_current+1;
            end
            vel_end = vel_face_T(i_current+1,j_current-1);
            vel_avg = (vel_end + vel_start)/2;
            dist_move = R_boundary(1,2)*(j_current-j_str)*delta_theta;
            time_move = dist_move/vel_avg;
            rad_exit = R_boundary(1,2);
            theta_exit = theta_boundaries (1,j_current);
                stream_line(count+1,1) = rad_exit;
                stream_line(count+1,2) = theta_exit;
                stream_line(count+1,3) = 0;
                stream_line(count+1,4) = 0;
                stream_line(count+1,5) = time_move;
                rad_exit = r_w;
                stream_line(count+2,1) = rad_exit;
                stream_line(count+2,2) = theta_exit;
                stream_line(count+2,3) = 0;
                stream_line(count+2,4) = 0;
                stream_line(count+2,5) = 0.0001;
        end
        if vel_face_T(i_current+1,j_current) > 0
            block_count = 0;
            j_str = j_current;
            vel_start = vel_face_T(i_current,j_current);
            while K_isotropic_blocks(i_current,j_current) ~= perf_perm
                block_count = block_count+1;
                j_current = j_current-1;
            end
            vel_end = vel_face_T(i_current+1,j_current+1);
            vel_avg = (vel_end + vel_start)/2;
            dist_move = R_boundary(1,2)*(j_current-j_str)*delta_theta;
            time_move = -1*dist_move/vel_avg;

            rad_exit = R_boundary(1,2);
            theta_exit = theta_boundaries (1,j_current);
                stream_line(count+1,1) = rad_exit;
                stream_line(count+1,2) = theta_exit;
                stream_line(count+1,3) = 0;
                stream_line(count+1,4) = 0;
                stream_line(count+1,5) = time_move;
                rad_exit = r_w;
                stream_line(count+2,1) = rad_exit;
                stream_line(count+2,2) = theta_exit;
                stream_line(count+2,3) = 0;
                stream_line(count+2,4) = 0;
                stream_line(count+2,5) = 0.0001;
        end
    end
end
```

Code Name: stremline_polluck_more_pts

Code Number: 19

```
% Basic streamline logic full
%----------DATA LOADING BEGIN----------
vel_field_coeff;                    % Velocity coefficients
%----------DATA LOADING ENDS----------
% creating starting points
total_pts = J * J_per_block;
rad_start_all = ones(1,total_pts)*r_e;
theta_start_all = ones(1,total_pts);
count_fill = 1;
while count_fill <= total_pts
    theta_start_all(1,count_fill) =
count_fill/total_pts*J*delta_theta+delta_theta/(J_per_block+1);
    count_fill = count_fill + 1;
end
count_fill = 1;
while count_fill <= total_pts
    stream_line_x(1,count_fill*2-1) = rad_start_all(1,count_fill);
    stream_line_x(1,count_fill*2) = theta_start_all(1,count_fill);%even
    count_fill = count_fill + 1;
end
% starting multiple streamlines
stream_counter = 1;
while stream_counter <= total_pts
    % setting entrance point
    stream_line = [rad_start_all(1,stream_counter),
theta_start_all(1,stream_counter)];
    theta_enter = stream_line(1,2);
    if theta_enter >= 2*pi()
        theta_enter = theta_enter-2*pi();
    end
    rad_enter = stream_line(1,1);
    % determining i and j locations
    j_current = ceil(stream_line(1,2)/delta_theta);
    if j_current == J+1
        j_current = 1;
    end
    i_current = N;
    rad_exit = stream_line(1,1)*0.99; % just for starting purposes
    count = 1;
    while rad_exit > r_w && rad_exit <= r_e
            % determining full sector distance travel times
            % determining theta dirn initial exit points
        if vel_face_T(i_current,j_current) > 0
            if j_current < J
                if vel_face_T(i_current,j_current+1) > 0 % case 1 -
exit cw
                    theta_exit = theta_boundaries(1,j_current);
                end
                if vel_face_T(i_current,j_current+1) < 0 % case 2 -
moves outward
```

```
                    theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) *
delta_theta/(vel_face_T(i_current,j_current+1)-
vel_face_T(i_current,j_current));
                    if theta_enter < theta_star
                        theta_exit = theta_boundaries(1,j_current);
                    else
                        theta_exit = theta_boundaries(1,j_current+1);
                    end
                end
                if vel_face_T(i_current,j_current+1) == 0 % case 1*
                    theta_exit = theta_boundaries(1,j_current);
                end
            end
            if j_current == 1 % speical case for boundary condition
                if vel_face_T(i_current,2) > 0 % case 1 - exit cw
                    theta_exit = theta_boundaries(1,j_current);
                end
            end
            if j_current == J % speical case for boundary condition
                if vel_face_T(i_current,1) > 0 % case 1 - exit cw
                    theta_exit = theta_boundaries(1,j_current);
                end
                if vel_face_T(i_current,1) < 0 % case 2 - moves outward
                    theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) * delta_theta/(vel_face_T(i_current,1)-
vel_face_T(i_current,j_current));
                    if theta_enter < theta_star
                        theta_exit = theta_boundaries(1,j_current);
                    else
                        theta_exit = theta_boundaries(1,1);
                    end
                end
            end
        end
        if vel_face_T(i_current,j_current) < 0
            if j_current < J
                if vel_face_T(i_current,j_current+1) < 0 % case 3 -
exit ccw
                    theta_exit = theta_boundaries(1,j_current+1);
                end
                if vel_face_T(i_current,j_current+1) > 0 % case 4 -
moves inward
                    theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) *
delta_theta/(vel_face_T(i_current,j_current+1)-
vel_face_T(i_current,j_current));
                    theta_exit = theta_star;
                end
            end
            if j_current == J % speical case for boundary condition
                if vel_face_T(i_current,1) < 0 % case 3 - exit ccw
                    theta_exit = 2*pi();  %theta_boundaries(1,1);
                end
                if vel_face_T(i_current,1) > 0 % case 4 - moves inward
```

```
                    theta_star = theta_boundaries(1,j_current) -
vel_face_T(i_current,j_current) * delta_theta/(vel_face_T(i_current,1)-
vel_face_T(i_current,j_current));
                    theta_exit = theta_star;
                end
            end
        end
        if vel_face_T(i_current,j_current) == 0 % radial case
            if j_current < J
                if vel_face_T(i_current,j_current+1) < 0 % case 3 -
exit ccw
                    theta_exit = theta_boundaries(1,j_current+1);
                end
                if vel_face_T(i_current,j_current+1) > 0 % case 4 -
moves inward
                    theta_exit = theta_boundaries(1,j_current);
                end
                if vel_face_T(i_current,j_current+1) == 0 % pure
radial
                    theta_exit = theta_enter;
                end
            end
            if j_current == J % special case for boundary condition
                if vel_face_T(i_current,1) < 0 % case 3 - exit ccw
                    theta_exit = theta_boundaries(1,1);
                end
                if vel_face_T(i_current,1) > 0 % case 4 - moves inward
                    theta_exit = theta_boundaries(1,j_current);
                end
            end
        end
        % determining theta dir'n travel time
        if theta_exit == theta_enter % radial
            time_theta = 1e50; % any large number (will look from min)
        else
            time_theta =
abs(1/vel_coeff_theta_a(i_current,j_current)*(log(vel_coeff_theta_b(i_c
urrent,j_current)+vel_coeff_theta_a(i_current,j_current)*theta_exit)-
log(vel_coeff_theta_b(i_current,j_current)+vel_coeff_theta_a(i_current,
j_current)*theta_enter)));
        end
        % determining radial dir'n travel time
        if abs(vel_coeff_rad_b(i_current,j_current)) > 1e-10
            time_rad =
1/(vel_coeff_rad_b(i_current,j_current)^2)*(vel_coeff_rad_b(i_current,j
_current)*(rad_enter-R_boundary(1,i_current))-
vel_coeff_rad_a(i_current,j_current)*(log(vel_coeff_rad_a(i_current,j_c
urrent)+vel_coeff_rad_b(i_current,j_current)*rad_enter) -
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*R_boundary(1,i_current))));
        end
        if abs(vel_coeff_rad_b(i_current,j_current)) < 1e-10 % no
leakage - radial
            time_rad =
1/(2*vel_coeff_rad_a(i_current,j_current))*(rad_enter^2-
R_boundary(1,i_current)^2);
        end
```

```
        % determining minimum travel time
        time_min = min(time_theta,time_rad);
        % converting to real time (from darcy time)
        % added matrix for flowrate calc
        flow_rate_data(count,1) = time_min * poro_const;
        flow_rate_data(count,2) = i_current;
        flow_rate_data(count,3) = j_current;
        % determining particle exit location
        if time_min == time_rad
            if abs(vel_coeff_theta_a(i_current,j_current)) < 1e-15
                theta_exit = theta_enter;
            end
            if abs(vel_coeff_theta_a(i_current,j_current)) > 1e-15
                theta_exit =
((vel_coeff_theta_b(i_current,j_current)+vel_coeff_theta_a(i_current,j_
current)*theta_enter)/exp(time_min*vel_coeff_theta_a(i_current,j_curren
t))-
vel_coeff_theta_b(i_current,j_current))/vel_coeff_theta_a(i_current,j_c
urrent);
            end
            rad_exit = R_boundary(1,i_current);
            i_current = i_current - 1;
            theta_enter = theta_exit;
            rad_enter = rad_exit;
        end
        if time_min == time_theta
            rad_exit_guess =  R_boundary(1,i_current);

            NR_num = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
rad_exit_guess)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(
i_current,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_
coeff_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*rad_exit_guess));
            NR_deno = (1/vel_coeff_rad_b(i_current,j_current)) -
(vel_coeff_rad_a(i_current,j_current)/vel_coeff_rad_b(i_current,j_curre
nt))*(1/(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current
,j_current)*rad_exit_guess));
            rad_exit = rad_exit_guess - (NR_num/NR_deno);
            while abs(rad_exit-rad_exit_guess)/abs(rad_exit) > 0.00001
                rad_exit_guess = rad_exit;
                NR_num = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
rad_exit_guess)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(
i_current,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_
coeff_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*rad_exit_guess));
                NR_deno = (1/vel_coeff_rad_b(i_current,j_current)) -
(vel_coeff_rad_a(i_current,j_current)/vel_coeff_rad_b(i_current,j_curre
nt))*(1/(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current
,j_current)*rad_exit_guess));
                rad_exit = rad_exit_guess - (NR_num/NR_deno);
            end
            if isreal(rad_exit) == 0
                % instead of NR, use
```

```
                % bisection-----------------------------------------
                guess_upper = R_boundary(1,i_current+1);
                guess_lower = R_boundary(1,i_current);
                answer_upper = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_upper)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_c
urrent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coe
ff_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_upper));
                answer_lower = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_lower)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_c
urrent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coe
ff_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_lower));
                product_guesses = answer_lower*answer_upper;
                if product_guesses > 0
                    fprintf('ERROR --> Bisection not bounding answer')
                end
                if product_guesses < 0
                    guess_new = (guess_upper+guess_lower)/2;
                    answer_new_guess = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_new)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_cur
rent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coeff
_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_new));
                end
                if answer_new_guess*answer_upper < 0
                    new_bound = guess_upper;
                end
                if answer_new_guess*answer_upper > 0
                    new_bound = guess_lower;
                end
                while new_bound-guess_new > 0.00001
                    answer_1 = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
new_bound)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_cur
rent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coeff
_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*new_bound));
                    answer_2 = time_min -
(1/vel_coeff_rad_b(i_current,j_current))*(rad_enter-
guess_new)+(vel_coeff_rad_a(i_current,j_current)/(vel_coeff_rad_b(i_cur
rent,j_current))^2)*(log(vel_coeff_rad_a(i_current,j_current)+vel_coeff
_rad_b(i_current,j_current)*rad_enter)-
log(vel_coeff_rad_a(i_current,j_current)+vel_coeff_rad_b(i_current,j_cu
rrent)*guess_new));
                    if answer_1*answer_2 < 0
                        new_bound = new_bound;
                    end
                    if answer_1*answer_2 > 0
                        new_bound = guess_new;
```

```
                    end
                    guess_new = (guess_new+new_bound)/2;
                end
                rad_exit = guess_new;
            end
% end bisection-----------------------------------------------------------
            if rad_exit < r_w
                rad_exit = r_w;
            end
            if theta_exit > theta_enter
                j_current = j_current + 1;
                if j_current == J+1
                    j_current = 1;
                end
            end
            if theta_exit < theta_enter
                j_current = j_current - 1;
                if j_current == 0
                    j_current = J;
                end
            end
            if theta_exit >= 2*pi()
                theta_exit = theta_exit-2*pi();
            end
% current addition to solve issue
            rad_enter = rad_exit;
            theta_enter = theta_exit;
        end
        if i_current > 0
            if vel_face_T(i_current,j_current) > 0
                if theta_exit == 0
                    theta_exit = 2*pi();
                    theta_enter = theta_exit;
                end
            end
        end
        flow_rate_data_x(count,stream_counter*3-2) =
flow_rate_data(count,1);
        flow_rate_data_x(count,stream_counter*3-1) =
flow_rate_data(count,2);
        flow_rate_data_x(count,stream_counter*3) =
flow_rate_data(count,3);
        count = count+1;
        stream_line(count,1) = rad_exit;
        stream_line(count,2) = theta_exit;
        stream_line_x(count,stream_counter*2-1) = stream_line(count,1);
        stream_line_x(count,stream_counter*2) = stream_line(count,2);
    %%% forcing casing hits to go to perf
    if rad_exit == R_boundary(1,2) &&
K_isotropic_blocks(i_current,j_current) == casing_perm
        if vel_face_T(i_current+1,j_current) < 0
            block_count = 0;
            j_str = j_current;
            vel_start = vel_face_T(i_current,j_current);
            while K_isotropic_blocks(i_current,j_current) ~= perf_perm
                block_count = block_count+1;
                j_current = j_current+1;
```

```
                if j_current == J+1
                    j_current = 0;
                end
            end
            vel_end = vel_face_T(i_current+1,j_current-1);
            vel_avg = (vel_end + vel_start)/2;
            dist_move = R_boundary(1,2)*(j_current-j_str)*delta_theta;
            time_move = abs(dist_move/vel_avg);
            rad_exit = R_boundary(1,2);
            theta_exit = theta_boundaries (1,j_current);
                stream_line(count+1,1) = rad_exit;
                stream_line(count+1,2) = theta_exit;
                stream_line_x(count+1,stream_counter*2-1) =
stream_line(count+1,1);
                stream_line_x(count+1,stream_counter*2) =
stream_line(count+1,2);
                flow_rate_data_x(count,stream_counter*3-2) = time_move*
poro_const;
                flow_rate_data_x(count,stream_counter*3-1) = i_current;
                flow_rate_data_x(count,stream_counter*3) = j_current;
                rad_exit = r_w;
                stream_line(count+2,1) = rad_exit;
                stream_line(count+2,2) = theta_exit;
                stream_line_x(count+2,stream_counter*2-1) =
stream_line(count+2,1);
                stream_line_x(count+2,stream_counter*2) =
stream_line(count+2,2);
                flow_rate_data_x(count+1,stream_counter*3-2) = 0.0001;
                flow_rate_data_x(count+1,stream_counter*3-1) =
i_current-1;
                flow_rate_data_x(count+1,stream_counter*3) = j_current;
        end

        if vel_face_T(i_current+1,j_current) > 0
            block_count = 0;
            j_str = j_current;
            vel_start = vel_face_T(i_current,j_current);
            while K_isotropic_blocks(i_current,j_current) ~= perf_perm
                block_count = block_count+1;
                j_current = j_current-1;
                if j_current == 0
                    j_current = J;
                end
            end
            vel_end = vel_face_T(i_current+1,j_current+1);
            vel_avg = (vel_end + vel_start)/2;
            dist_move = R_boundary(1,2)*(j_current-j_str)*delta_theta;
            time_move = abs(dist_move/vel_avg);
            rad_exit = R_boundary(1,2);
            theta_exit = theta_boundaries (1,j_current);
                stream_line(count+1,1) = rad_exit;
                stream_line(count+1,2) = theta_exit;
                stream_line_x(count+1,stream_counter*2-1) =
stream_line(count+1,1);
                stream_line_x(count+1,stream_counter*2) =
stream_line(count+1,2);
```

```
                flow_rate_data_x(count,stream_counter*3-2) = time_move*
poro_const;
                flow_rate_data_x(count,stream_counter*3-1) = i_current;
                flow_rate_data_x(count,stream_counter*3) = j_current;
                rad_exit = r_w;
                stream_line(count+2,1) = rad_exit;
                stream_line(count+2,2) = theta_exit;
                stream_line_x(count+2,stream_counter*2-1) =
stream_line(count+2,1);
                stream_line_x(count+2,stream_counter*2) =
stream_line(count+2,2);
                flow_rate_data_x(count+1,stream_counter*3-2) = 0.0001;
                flow_rate_data_x(count+1,stream_counter*3-1) =
i_current-1;
                flow_rate_data_x(count+1,stream_counter*3) = j_current;
        end
    end
    end
    count = 1;
    stream_counter = stream_counter + 1;
end
```

Code Name: upscaled_mobility

Code Number: 20

```
% File for upscaled transmissibility's
% Note i,j corresponds with row, column
%----------DATA LOADING BEGIN----------
input_data;                             % Input data file
permeability;                           % Node permeability matrix
distances;                              % Node spacing
%----------DATA LOADING ENDS----------
mobility_isotropic_blocks = K_isotropic_blocks/mu_o;
% Creating upscaled radial mobility matrix
% Note there is one more upscaled value than block values
mobility_upscale_R=ones(N+1,J);
mobility_upscale_R(1,:) = mobility_isotropic_blocks(1,:);
mobility_upscale_R(N+1,:) = mobility_isotropic_blocks(N,:);
i = 2;
j = 1;
while i <= N
    while j <= J
        mobility_upscale_R(i,j) = log(R_nodes(1,i)/R_nodes(1,i-
1))/((1/mobility_isotropic_blocks(i-
1,j)*log(R_boundary(1,i)/R_nodes(1,i-
1)))+(1/mobility_isotropic_blocks(i,j))*log(R_nodes(1,i)/R_boundary(1,i
)));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% creating upscaled tangential mobility matrix
mobility_upscale_T = ones(N,J);
i = 1;
j = 1;
while i <= N
    while j < J
        mobility_upscale_T(i,j) = (2 * mobility_isotropic_blocks(i,j) *
mobility_isotropic_blocks(i,j+1)) / (mobility_isotropic_blocks(i,j) +
mobility_isotropic_blocks(i,j+1));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
% fixing end condition (first node to last node at theta = 0)
i=1;
while i <= N
    mobility_upscale_T(i,J)= (2 * mobility_isotropic_blocks(i,1) *
mobility_isotropic_blocks(i,J)) / (mobility_isotropic_blocks(i,1) +
mobility_isotropic_blocks(i,J));
    i = i+1;
end
```

Code Name: vel_face_plot_R

Code Number: 21

```
% creating face velocity plotting file - radial
%----------DATA LOADING BEGIN----------
face_velocities;                    % face velocities
%----------DATA LOADING ENDS----------
% determining x,y coordinates of nodes
nodes_theta = linspace(2*pi/(2*J),2*pi-2*pi/(2*J),J);
% note boundary location point are in line with nodes
boundary_x = zeros(N,J);
boundary_y = zeros(N,J);
i = 1;
j = 1;
while i <= N+1;
   while j <= J;
     boundary_x(i,j) = R_boundary(1,i)*cos(nodes_theta(1,j));
     boundary_y(i,j) = R_boundary(1,i)*sin(nodes_theta(1,j));
     j = j+1;
   end
   i = i+1;
   j = 1;
end
% creating 3D x,y,z plot
surf(boundary_x,boundary_y,vel_face_R)
```

Code Name: vel_face_plot_T

Code Number: 22

```
% creating face velocity plotting file - tangential
%----------DATA LOADING BEGIN----------
face_velocities;                    % face velocities
%----------DATA LOADING ENDS----------
% determining x,y corridinates of nodes
nodes_theta_tanj_bound = linspace(0,2*pi,J);
% note boundary location point are in line with nodes
boundary_x_tanj = zeros(N,J);
boundary_y_tanj = zeros(N,J);
i = 1;
j = 1;
while i <= N;
   while j <= J;
     boundary_x_tanj(i,j) =
R_boundary(1,i)*cos(nodes_theta_tanj_bound(1,j));
     boundary_y_tanj(i,j) =
R_boundary(1,i)*sin(nodes_theta_tanj_bound(1,j));
     j = j+1;
   end
   i = i+1;
   j = 1;
end
% creating 3D x,y,z plot
surf(boundary_x_tanj,boundary_y_tanj,vel_face_T)
```

Code Name: vel_face_T_fix

Code Number: 23

```
% face velocity fix
% this file will search through the face velocity matrixes to delete
% velocities smaller than abs(1e-12( and replace them with 0 - meaning
% no velocity along that face
i = 1;
j = 1;
while i <= N
    while j<=J
        if abs(vel_face_T(i,j)) < 1e-12
            vel_face_T(i,j) = 0;
        end
        j=j+1;
    end
    i=i+1;
    j=1;
end
```

Code Name: vel_field_coeff

Code Number: 24

```
% determining velocity field coefficients
%----------DATA LOADING BEGIN----------
face_velocities;                % Face velocities
vel_face_T_fix;                 % Fix face velocities
%----------DATA LOADING ENDS----------
% first creating theta values for boundaries
theta_boundaries = ones(1,J);
j = 0;
while j < J
    theta_boundaries(1,j+1) = j*delta_theta;
    j = j+1;
end
% solving for unknown velocity field coeff in the theta direction
% (a-theta and b-theta)
vel_coeff_theta_a = ones(N,J);
vel_coeff_theta_b = ones(N,J);
vel_coeff_rad_a = ones(N,J);
vel_coeff_rad_b = ones(N,J);
i = 1;
j = 1;
while i <= N
    while j < J
        vel_coeff_theta_a(i,j) = (vel_face_T(i,j) - vel_face_T(i,j+1))
/ (theta_boundaries(1,j) - theta_boundaries(1,j+1));
        vel_coeff_theta_b(i,j) = vel_face_T(i,j+1) -
(vel_coeff_theta_a(i,j)*theta_boundaries(1,j+1));
        j = j+1;
    end
    if j == J
        vel_coeff_theta_a(i,j) = (vel_face_T(i,j) - vel_face_T(i,1)) /
(theta_boundaries(1,j) - theta_boundaries(1,1));
        vel_coeff_theta_b(i,j) = vel_face_T(i,1) -
(vel_coeff_theta_a(i,j)*theta_boundaries(1,1));
    end
    i = i+1;
    j = 1;
end
i = 1;
j = 1;
while i <= N
    while j <= J
        vel_coeff_rad_a(i,j) =
R_boundary(1,i)*R_boundary(1,i+1)/(R_boundary(1,i+1) - R_boundary(1,i))
* (vel_face_R(i,j)-vel_face_R(i+1,j));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
i = 1;
```

```
j = 1;
while i <= N
    while j <= J
        vel_coeff_rad_b(i,j) = vel_face_R(i+1,j)-
(vel_coeff_rad_a(i,j)/R_boundary(1,i+1));
        j = j+1;
    end
    i = i+1;
    j = 1;
end
```

Appendix II - Case Study 1: Fractional Factorial Run Results

Run	Factor A	Factor B	Factor C	Factor D	Factor E	Factor F	Response 1	Response 2
	Damage Zone Radius (m)	Damage Zone Permeability (mD)	Perforation Length (m)	Perforation Thickness (m)	Perforation Permeability (mD)	Number of Perforations	Skin	Flow Rate (bbl/d)
1	0.1	90	0.15	0.03	100	3	1.1	536
2	0.5	15	0.75	0.03	100	1	23.0	143
3	0.5	90	0.75	0.03	100	3	1.2	526
4	0.1	90	0.75	0.05	1600	1	0.0	620
5	0.5	15	0.15	0.03	1600	1	18.7	166
6	0.5	15	0.75	0.03	1600	3	0.0	619
7	0.1	90	0.75	0.05	100	3	0.5	574
8	0.1	15	0.15	0.03	100	1	11.2	235
9	0.5	90	0.15	0.05	100	3	0.7	562
10	0.5	15	0.15	0.03	100	3	11.3	235
11	0.5	15	0.75	0.05	1600	1	3.1	427
12	0.5	90	0.15	0.03	1600	3	-0.3	649
13	0.5	15	0.15	0.05	1600	3	7.8	289
14	0.5	90	0.75	0.05	100	1	4.1	387
15	0.1	90	0.15	0.05	1600	3	-0.6	682
16	0.1	15	0.75	0.05	100	1	7.7	292
17	0.5	15	0.15	0.05	100	1	21.4	151
18	0.5	90	0.75	0.03	1600	1	0.7	559
19	0.1	90	0.75	0.03	1600	3	-1.2	748
20	0.5	90	0.15	0.03	100	1	5.8	336
21	0.1	15	0.15	0.05	100	3	1.5	505
22	0.1	90	0.15	0.03	1600	1	1.5	511

23	0.1	15	0.75	0.05	1600	3	-1.5	787
24	0.5	15	0.75	0.05	100	3	6.8	312
25	0.1	15	0.15	0.03	1600	3	-0.3	647
26	0.1	90	0.15	0.05	100	1	3.9	395
27	0.1	15	0.75	0.03	100	3	2.7	444
28	0.1	15	0.15	0.05	1600	1	1.9	485
29	0.1	15	0.75	0.03	1600	1	1.2	524
30	0.5	90	0.75	0.05	1600	3	-1.5	787
31	0.1	90	0.75	0.03	100	1	5.6	342
32	0.5	90	0.15	0.05	1600	1	1.2	525
33	0.3	52.5	0.45	0.04	850	1	2.1	475
34	0.3	52.5	0.45	0.04	850	3	-0.6	674

Appendix III - Case Study 2: IV Optimal Design Run Results

Run	Factor A	Factor B	Factor C	Response 1	Response 2
	Perforation Length (m)	Perforation Permeability (mD)	Number of Perforations	Total Skin	Flow Rate (bbl/d)
1	0.21375	1480	2	0.9	546
2	0.7025	1075	1	2.1	476
3	0.1755	1105	1	3.8	399
4	0.43475	1105	2	0.2	597
5	0.58775	1600	2	-0.4	652
6	0.6005	100	1	8.0	287
7	0.2945	850	3	0.3	596
8	0.413006007	595	1	3.4	413
9	0.8555	820	3	-0.4	660
10	0.85975	1337.5	2	-0.3	651
11	1	1600	1	1.3	519
12	0.15	370	3	1.5	511
13	0.38375	1472.5	1	2.2	466
14	0.15	1600	1	4.1	389
15	0.15	100	2	3.7	402
16	1	1307.5	3	-1.0	717
17	1	1150	1	1.9	486
18	0.83425	100	3	2.4	458
19	0.57925	1330	3	-0.8	697
20	0.847	1585	1	1.4	516
21	0.306976687	100	3	2.5	456
22	1	850	2	0.2	603
23	0.15	662.5	1	4.6	370
24	0.936269894	197.5	2	2.3	462
25	0.575	355	3	0.6	567
26	0.881	433.4986232	1	3.8	397
27	0.847	1600	3	-1.1	736
28	0.15	842.5	2	1.7	495
29	0.18825	220	1	5.9	334
30	0.56225	100	2	3.6	407
31	0.2945	362.5	2	1.7	498
32	0.15	1600	1	4.1	389
33	0.6175	1495	1	1.6	502
34	0.6855	1022.5	2	0.0	618
35	0.32	1600	3	-0.2	639
36	0.482715107	662.5	2	0.7	563
37	0.73225	587.5	2	0.7	562
38	1	100	1	8.0	287
39	1	355	3	0.6	570
40	0.4305	595	3	0.2	604
41	0.15	1330	3	1.0	541
42	0.6175	100	3	2.4	458
43	0.17125	1600	3	0.7	558

44	1	340	2	1.5	510
45	0.66	985	3	-0.6	674
46	0.24775	775	2	1.1	534
47	1	1600	3	-1.1	741
48	0.673850554	310	1	4.6	370
49	0.813	392.5	3	0.5	579
50	0.91925	1600	2	-0.6	672
51	0.28175	1300	3	0.0	614
52	1	737.5	3	-0.3	649
53	0.15	100	3	2.6	449
54	0.49	1600	1	1.7	496
55	1	625	1	3.0	429
56	1	1450	2	-0.5	662
57	0.32	1600	2	0.3	591
58	0.40925	250	1	5.3	351
59	0.15	325	2	2.3	465
60	0.15	1600	2	1.5	506
61	0.26475	775	1	3.4	413
62	0.5495	880	1	2.5	454
63	1	100	2	3.6	407

Appendix IV - Case Study 3: IV Optimal Design Run Results

Run	Factor A	Factor B	Factor C	Response 1	Response 3
	Filter Cake Permeability (mD)	Filter Cake Thickness (mm)	Angle of Coverage (degrees)	Total Skin	Flow Rate (bbl/d)
1	165.8	1.16	270.0	1.422	512
2	68.7	8.84	96.8	-0.006	618
3	174.4	9.00	121.5	0.100	609
4	10.0	1.20	45.0	-0.154	631
5	10.0	9.00	50.6	-0.143	630
6	173.4	7.40	264.4	1.563	503
7	17.6	1.92	168.8	0.412	583
8	200.0	3.60	270.0	1.600	501
9	122.1	8.24	196.9	0.654	564
10	143.0	1.19	135.0	0.159	603
11	24.3	9.00	216.0	0.876	548
12	200.0	9.00	225.0	0.970	541
13	165.8	3.16	220.5	0.886	547
14	10.0	8.40	270.0	1.744	493
15	180.1	4.60	45.0	-0.154	631
16	200.0	7.84	61.9	-0.118	628
17	44.2	6.64	45.0	-0.153	631
18	46.1	1.00	83.1	-0.060	623
19	181.0	1.00	45.0	-0.155	632
20	115.5	4.72	270.0	1.680	497
21	80.2	1.80	146.3	0.241	597
22	10.0	1.00	264.4	1.601	501
23	137.3	6.52	78.8	-0.071	624
24	133.5	9.00	45.0	-0.154	631
25	37.6	2.44	270.0	1.702	495
26	15.7	3.92	77.6	-0.074	624
27	76.5	4.60	155.3	0.302	592
28	10.0	4.86	231.8	1.080	534
29	200.0	1.00	201.4	0.595	568
30	12.9	7.44	138.4	0.198	600
31	194.9	2.52	99.0	-0.003	618
32	101.1	2.44	51.8	-0.143	630
33	197.2	5.96	168.8	0.401	584
34	50.0	7.16	249.8	1.346	517
35	84.1	1.20	227.3	0.931	544
36	103.1	9.00	270.0	1.711	495