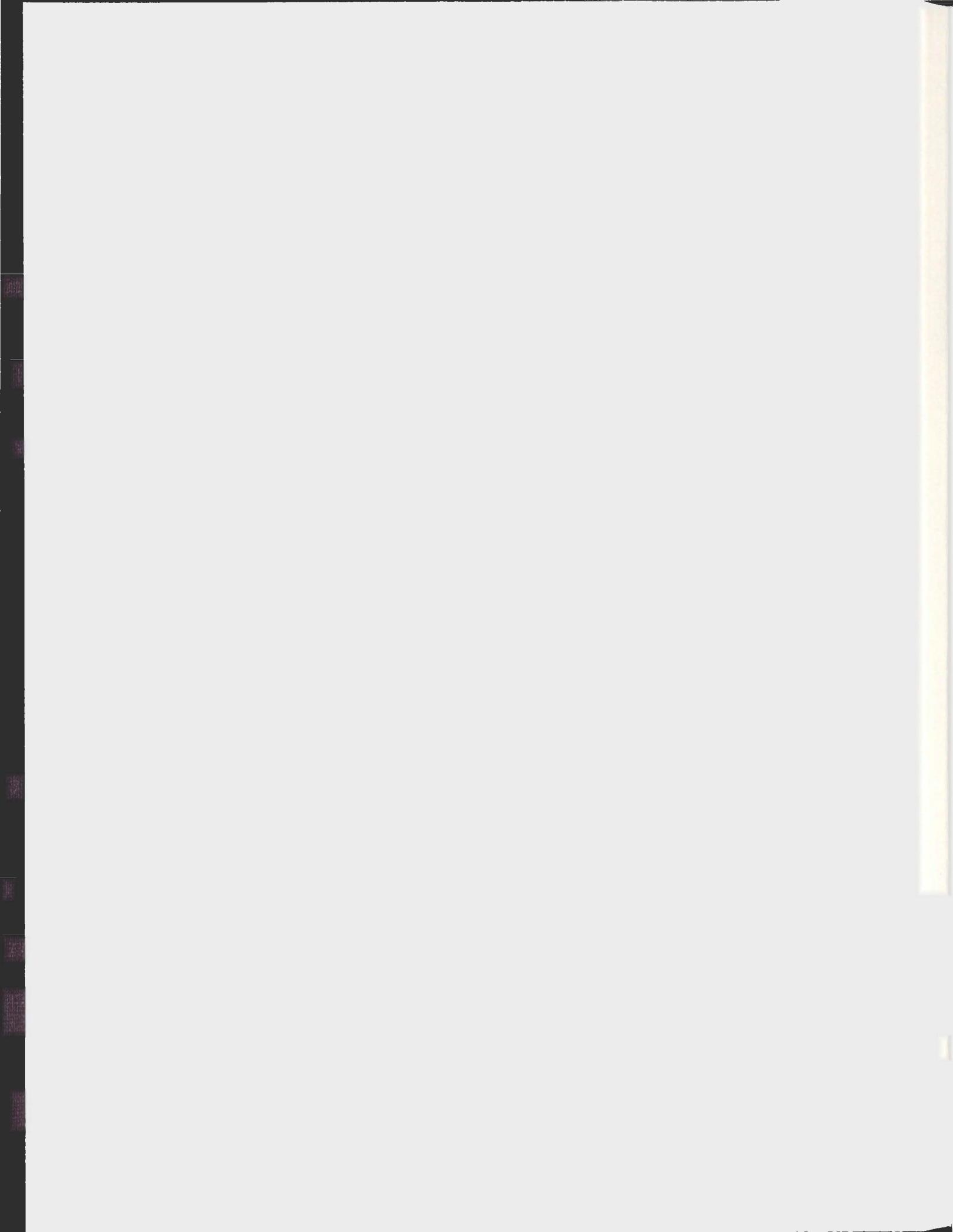


NEAR-WELL STREAMLINE SIMULATION UTILIZING
A CO-AXIAL CYLINDRICAL GRID STRUCTURE

JUSTIN C. PITTMAN





Near-Well Streamline Simulation Utilizing a Co-axial Cylindrical Grid Structure

BY

JUSTIN C. PITTMAN

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

DEPARTMENT OF EARTH SCIENCE

MEMORIAL UNIVERSITY OF NEWFOUNDLAND

SEPTEMBER 2012

ST. JOHN'S

NEWFOUNDLAND

Abstract

Accurate reservoir simulation is key in attaining an effective and efficient reservoir completion. Typical reservoir simulators employ grid sizes that are several orders of magnitude larger than the wellbore diameter and therefore, can not capture the near-well flow attributes. To improve the depiction of near-well flow phenomenon, a cylindrical grid system with a logarithmic radial-spacing is utilized; oriented such that small radial steps are taken in nearer regions allowing us to portray a more accurate flow description nearer to the wellbore. Streamline simulation techniques are the primary focus for this work as they promise computational efficiency and enhance flow visualization.

Herein, a robust streamline-based, cylindrical flow simulation method for the near-well region is presented. This will allow for a more accurate visualization of the near-well flow characteristics and improve overall reservoir flow simulation.

In this work, the streamline model is an adaptation of the semi-analytical method of [Pollock, 1988]; derived for a two-dimensional, cylindrical, single-phase, incompressible, homogeneous medium, for both isotropic and anisotropic permeability fields. A discretization scheme is also presented utilizing a novel logarithmically spaced grid for the numerical model, and tested for an isotropic and anisotropic case for illustration. The results show that this methodology has potential to be useful in determining the flow characteristics in the near-well region and provide stable results.

Acknowledgments

Dedicated to the memory of my father, Edgar Maxwell Pittman, whose teachings and support will never be forgotten.

Completing this M.Sc. has been a marathon event. One in which I would not have been able to finish without encouragement. A special thanks must go to my love, Jennifer, whom gave birth to our first son, Jakob, in the course of this research. Thank you. To my supervisor, Dr. Thormod Johansen, for his assistance, understanding, and patience throughout this work; I must extend the utmost of appreciation for the opportunity and guidance.

This has been a work of much interest, hard work, and frustration. The sheer amount of focus and diligence that is demanded of a graduate student is remarkable and should be recognized in its own right; it's quite the undertaking. I've had tremendous support from family and friends, and to them I am grateful.

Contents

1	Introduction	1
1.1	Media properties	2
1.1.1	Porosity	3
1.1.2	Permeability	4
1.2	Pathlines, streamlines, and streaklines	5
1.3	Streamline simulation in near-well	8
1.4	Scope of this thesis	9
2	Mathematical Model	13
2.1	Incompressible single-phase flow	13
2.2	Streamfunctions and streamline tracing	18
2.2.1	Classical streamfunctions	19
2.2.1.1	Isotropic streamfunction in two dimensions	19
2.2.1.2	Anisotropic streamfunction in two dimensions	23
2.2.2	Streamfunctions in three-dimensions	25
2.3	Streamline tracing and particle tracking techniques	27
2.3.1	Streamline tracing	28
2.3.2	Pollock's method in 2D Cartesian	29
2.4	Literature review	32

3	Streamline Tracing in Cylindrical Coordinates	38
3.1	Cylindrical transform	39
3.2	Cylindrical velocity interpolation scheme	43
3.2.1	Radial velocity interpolation	43
3.2.2	Angular velocity interpolation	45
3.2.3	TOF and radial displacement	46
3.2.4	TOF and the angular displacement	49
3.3	Steady-state flow and the TOF	51
4	Discretization For Implementation	53
4.1	Discretization for pressure simulation	55
4.2	Discretization Code in Matlab	57
4.2.1	Isotropic medium	57
4.2.2	Anisotropic medium	58
4.3	Radial grid discretization	59
4.4	Cylindrical velocity discretization	63
4.4.1	Numerical gradients	64
4.4.2	Upscaling procedures	65
4.4.3	Discretized flow velocity expressions	67
4.5	Discretized TOF expressions	69
4.5.1	Radial TOF	70
4.5.2	Angular TOF	70
4.6	Exit point determination	71
4.6.1	Radial exit point	72
4.6.2	Angular exit point	73
4.7	General code structure	73

5	Results and Conclusions	75
5.1	Pressure calculation for quarter five-spot configuration	78
5.1.1	Isotropic pressure calculation	79
5.1.2	Anisotropic pressure calculation	83
5.2	Streamline results	84
5.2.1	Isotropic Streamline	86
5.2.2	Anisotropic streamline	89
5.3	Future Work	95
5.4	Concluding remarks	97
A	TPFA for Pressure Calculation	99
B	Radial Grid Setup Matlab Code	102
C	Streamline Simulation for Radial Grid Code	112

List of Figures

1.1	The plot shows fluid flow under the velocity field $\mathbf{u}(x, y, t) = (1, yt)$ where time advances in each plot, from <i>a</i>) to <i>d</i>), clockwise. The red dot represents a particle released from the origin at time $t = 0$, tracing out its <i>pathline</i> in red. As it moves, it leaves behind a trail of blue ink (starting from the origin), which represents its <i>streakline</i> . The velocity is represented twice, with flow vectors, and a family of <i>streamlines</i> which are entirely redefined with each time step due to the changing flow [FilKaiv8, 2010].	6
1.2	Areal view of the cylindrical grid structure to be overlaid co-axially with a wellbore model designed to simulate the flow trajectories of the near-well region. Streamline techniques will be defined to illustrate the flow of the constituent fluid as it navigates a simulated structure. . .	11
2.1	The stream-surfaces $\psi(x, y, z)$ and $\chi(x, y, z)$ intersect at the streamlines, which are tangent to the averaged fluid-velocity vector. The cross-sectional area bounded by the stream-surfaces - shaded above - are mass-conservative [Bear, 1972].	26

2.2	Illustration demonstrating the Pollock methodology for a single Cartesian block. Here the face velocities for each side is show by u_{x1} , u_{x2} , u_{y1} , and u_{y2} . As well, the pathline and entry/exit points through the block are shown.	29
3.1	Radial annular region. Here we have a illustration of an annular region of the cylindrical grid-block. We note the radial and angular velocity values and arrows associated with an arbitrary flow direction; if flow is found to not be in the direction chosen, the associated values will be negative. An imaginary particle, in the case shown here, enters at the block at the point (r_p, θ_p, t_p) signifying a radius r_p , an angle θ_p , at a time value t_p . As the particle passes through the block, determined based on the face velocities, u_{r1} , u_{r2} , $u_{\theta1}$, and $u_{\theta2}$, exits the block at (r_e, θ_e) at time t_e	47
4.1	Pressure contours for simulated region. Red colour indicates relative high pressure, blue relative low. Plot <i>a</i>) shows the pressure contours for the first quadrant of a simulated anisotropic porous medium with a flow sink at block (1,1) with the node locations plotted. Plot <i>b</i>) are the pressure contours constructed using only the cylindrical node pressure values from the square Cartesian system. Plot <i>c</i>) is the node data plotted as a Cartesian array. The detailed code used to produce these plots is provided in Appendix A and B.	54
4.2	This is the Matlab code employed to set up the grid dimensions and flow conditions for the isotropic case. We note that the last line in the code calls upon the <i>TPFA2</i> function that we defined in the previous section.	58

- 4.3 Result of having isotropic permeability on the pressure for TPFA simulation in Matlab with a sink placed near the center of the grid system. Plot *a)* illustrates the grid layout as the top layer and the projected pressure contours on the lower. Plot *b)* shows the isolated contour plot of the pressure distribution around the fluid sink mimicking the pressure drop due to a production well. 59
- 4.4 This is the Matlab code for the anisotropic case. Here, we note the assembling of the permeability matrix denoted *Grid.K* to define the anisotropy in the simulation. 60
- 4.5 Result of anisotropic permeability on pressure contours using TPFA simulation in Matlab. A permeability relation of $(0.6)K_x = K_y$ was chosen for the simulation which is reflected in that the major axis of the outer pressure ellipses are stretched in the *x*-direction appropriately. 61
- 4.6 Plot *a)* is radius versus number of points illustrating the radius values produced by the algorithm $r_{n+1} = r_n + r_n \Delta \theta$ properly space the radial steps according to the chosen initial radius and angular width of the blocks. Plot *b)* shows that a linear plot is produced by plotting of the natural log of radius vs the number of points, illustrating that the radii values are logarithmic. 62
- 4.7 Areal view of a section of the cylindrical grid structure illustrating the numbering of grid blocks. Nodes are designated (i, j) where *i* is the radial location and *j* is the angular location. The interface locations are marked here with their respective $1/2$ increments. 63

4.8	The identities of the sides of each grid block in the cylindrical block. The radius r_1 is the outermost boundary and r_2 the innermost. The angle $a_1 > a_2$ measured in the usual counter-clockwise direction. The velocities u are noted with the subscript identifying their respective face.	67
4.9	Flow chart showing the logic structure for streamline generation. . .	74
5.1	Isotropic pressure contours with radial grid overlay. Note the injection site at the top right, noted by a high pressure in red, and the production site in the bottom left corner where the colour turns blue denoting a low pressure. The radial grid system is overlaid illustrating the block dimensions.	79
5.2	Anisotropic pressure contours with radial grid overlay.	83
5.3	Plot of radial grid with start location and velocity vectors. Plot <i>a</i>) is of the entire grid with start location circled. This point is used for the beginning location of the test of the discretized code in the isotropic and anisotropic case. Plot <i>b</i>) is a close-up of the starting location and the first grid block. The velocities u_{r1} , u_{r2} , u_{a1} , and u_{a2} are illustrated.	85
5.4	For an isotropic system. Plot <i>a</i>) shows the streamline originating from outer radius and traveling towards the wellbore. Plot <i>b</i>) is a close-up of the streamline's trajectory moving roughly orthogonal to the radius contours.	86
5.5	For an anisotropic system. Plot <i>a</i>) shows the streamline originating from an-outer radius and travels toward the wellbore, but with a distinct curvature. Plot <i>b</i>) is a close-up of the streamline's initial trajectory illustrating an influence due to a more prominent angular flow.	90

Chapter 1

Introduction

Modern reservoir simulation is the culmination of decades employing numerical analysis techniques to solve a complex system of equations, attempting to capture a glimpse of the reality of flow in the subsurface. The theory of fluid flow in porous media – e.g. the earth’s subsurface – extends, in literature, to the roots of thermodynamics; much of the mathematical construct of fluid-flow in porous media is equivalently mathematically described by the condition of heat conduction, electrical conduction, mass diffusion, etc. Generally speaking, described by the equations of potential flow [Vanfai *et al*, 2005].

Efficient and effective hydrocarbon reservoir completion is contingent on accurate simulation of the flow characteristics. Current reservoir simulators approximate flow of the entire reservoirs which, in some cases, have dimensions extending kilometers with very complex geological structure. Despite advances in computational power available, certain approximations are required to allow current technologies to process such a system as it remains impossible to capture all of the flow characteristics - and will remain so for quite some time to come.

The methodology of reservoir simulation typically involves constructing a highly

complex grid structure based on knowledge of the geological subsurface, such that a finite difference approximation can be implemented to approximate the fluid flux through the grid cells. These grid systems often employ networks in excess of a million grid-cells with block sizes that are several orders of magnitude larger than that of the physical wellbore, from which fluids are expected to be extracted. A critical portion of the reservoir simulation takes place in the simulation of near-well region of the wellbore.

1.1 Media properties

Firstly, we must define some of the common terminology with which this document will refer. Although, at many points, this document refers to *porous media*, we are in fact referring to the subsurface of the earth's crust, mostly pertaining to how fluid is transported through the medium; specifically, application to hydrocarbon reservoir flow is the intention, but this work may apply equally to other fields, such as contaminant flow.

We understand that the medium through which most fluids in the subsurface flow are generally composed of layers of sediment that have collected over millions of years - a few centimeters every hundred years or so - where the temperature and pressure have created this substrate. Over that period, tectonic and volcanic activity have transformed these relatively smooth layers of sediment into complex structure of various rock types with varying physical properties. It is among these layers that organic material is trapped and under particular pressures and temperatures is turned to hydrocarbons where gravity then separates the fluid into components of varying densities. The lighter hydrocarbons such as methane generally escape the subsurface quickly, while the more dense tend to flow much more slowly. Now, among these

layers of sediment, certain rocks types create bent layers where fluid can no longer move and instead become trapped. These are the oil reservoirs that we have come to utilize.

It is also important to recognize that we are speaking of layers of solid rock, of which there are small interconnected pores that reservoir fluids are able to flow through. There are certain parameters of these rocks types that dictate the flow regime of the fluid in the medium.

1.1.1 Porosity

The rock *porosity*, denoted here as ϕ , is the volume fraction of the void space within the matrix structure of the medium. In other words, this value is a measure of the percentage of fluid volume that a given rock type can withhold; theoretically ranging from $0 \leq \phi \leq 1$, but in general is a value between 0 – 0.4.

In most texts, the porosity is defined also in terms of the compressibility of the medium, which is dependent on the pressure. The *compressibility* is defined as

$$c = \frac{1}{\phi} \frac{d\phi}{dP}$$

where P is the reservoir pore pressure. In most simplified reservoir models, the compressibility is neglected and instead the porosity ϕ is taken to be dependent only on the spatial coordinates.

In this text, when we speak of porosity, we will most often be referring to the *effective porosity* of the medium, which differs from the intrinsic porosity in that the effective porosity is an estimate of the interconnected pores that will allow transport of the constituent fluid within the medium. This is an important designation as, for our purposes, we are only concerned with the pore space within the medium where

transport is supported.

1.1.2 Permeability

The *permeability*, denoted by \mathbf{K} , is a measure of the ability for a medium to transport a fluid through its structure matrix. The permeability does not generally have any correlation with the porosity of a medium as it depends entirely on the connectedness of the pore structure. For instance, a medium may have a high porosity value, but unless these pores are interconnected, it may have little to no permeability and, hence, no fluid will flow.

In certain rocks types, such as sandstone, fairly uniform particles are compressed to produce the rock matrix and tend to have a very high permeability due to the relatively large interconnected pores. Other rocks, such as shales, which are composed of mud, clay minerals, and tiny silt-sized particles, are generally considered to be non-permeable, however, may have an intrinsic porosity value up to 0.1.

The S.I. unit of permeability is meters squared (m^2), however it is also commonly represented in *Darcy* (D) or in *milli-Darcy* (mD). The Darcy unit is defined as $1D \approx 0.987 \cdot 10^{-12} m^2$ where a value of $1D$ would be considered to be a relatively high permeability value.

The permeability, \mathbf{K} , is generally a *tensor* where the value is different for different directions and the value is dependent on other directions. It is usually possible or sufficient to reduce the \mathbf{K} tensor's components to its values into orthogonal spatial directions by diagonalizing the tensor as

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \simeq \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix}$$

for a Cartesian (x, y, z) space. The main permeability directions (x, y, z) are typically chosen to align to the geological structure, so as to capture the dominant natural flow. Here, we've expressed the tensor in matrix form for clarity.

The number of terms required in \mathbf{K} varies depending on the mathematical model required to capture the flow characteristics. In some cases, it is sufficient to reduce the tensor to a single value where $\mathbf{K}=\mathbf{K}$; this scenario is regarded as an *isotropic* porous medium, as the flow is generally similar in all directions. Although, in many cases this would be an irresponsible approximation. When more terms are necessary, say $\mathbf{K} = (K_x, K_y, K_z)$ denoting the permeability in the x , y , and z components, this is regarded as an *anisotropic* medium. A *generally anisotropic* medium will contain all nine components. In this study, we will consider an anisotropic medium as any permeability that requires more than one component.

1.2 Pathlines, streamlines, and streaklines

We will now introduce the notion of a streamline in fluid flow and we must be distinct in how we define the streamline. It is important to understand that there are several similar concepts to tracing the fluid flow, but each is defined in a unique way.

A *pathline* of a fluid particle is a tracing of the positions in space that it will follow as time passes. It is, specifically, the Lagrangian trajectory of an imaginary particle where we are able to map its spatial locations during its path as it flows with the fluid. It is described by three parametric equations

$$\frac{dx}{u_x(x, y, z, t)} = \frac{dy}{u_y(x, y, z, t)} = \frac{dz}{u_z(x, y, z, t)} = dt, \quad (1.1)$$

where u_x , u_y , and u_z are the components of the velocity vector \mathbf{u} . The velocity vectors

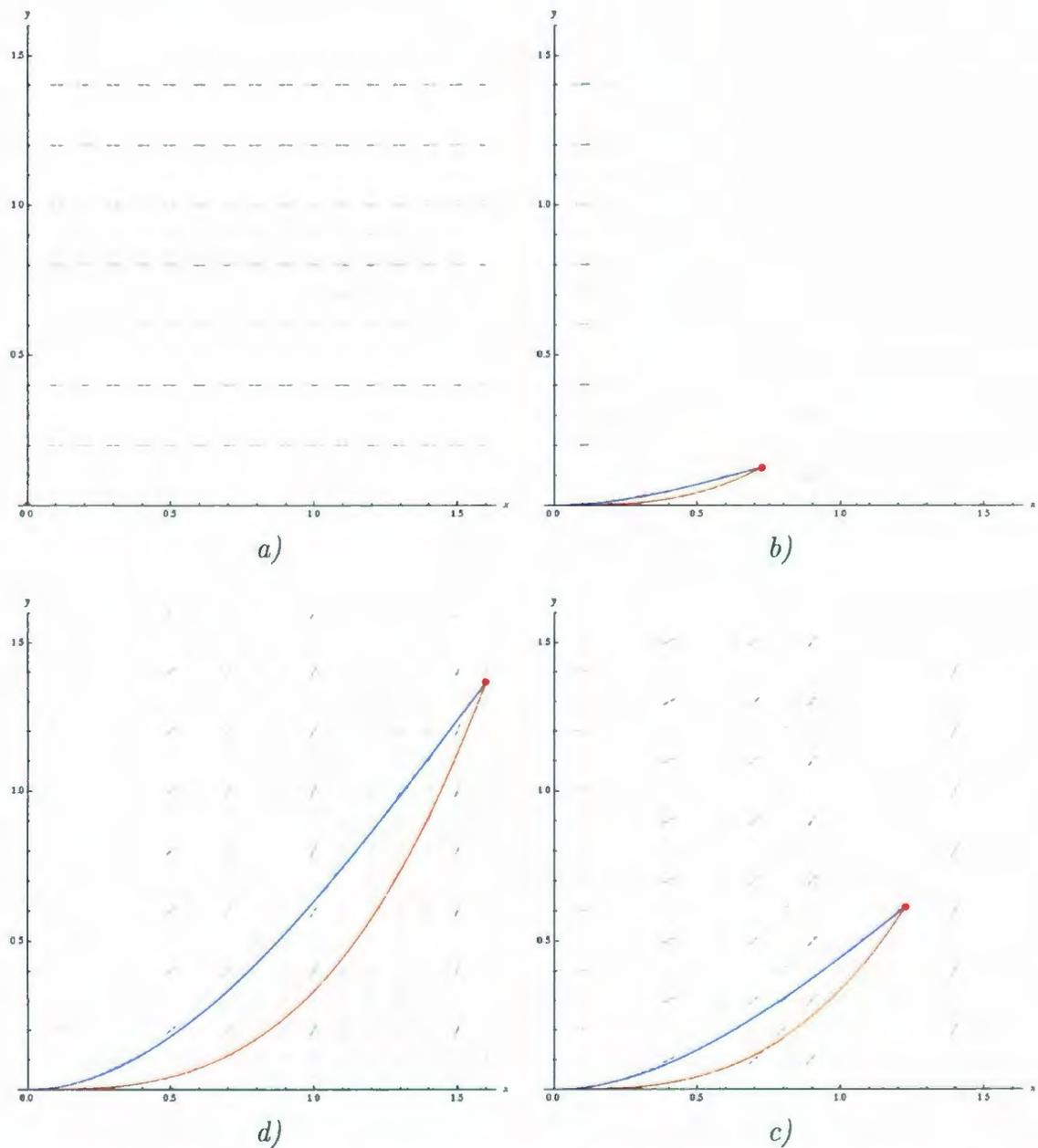


Figure 1.1: The plot shows fluid flow under the velocity field $\mathbf{u}(x, y, t) = (1, yt)$ where time advances in each plot, from a) to d), clockwise. The red dot represents a particle released from the origin at time $t = 0$, tracing out its *pathline* in red. As it moves, it leaves behind a trail of blue ink (starting from the origin), which represents its *streakline*. The velocity is represented twice, with flow vectors, and a family of *streamlines* which are entirely redefined with each time step due to the changing flow [Fi1Kaiv8, 2010].

here are defined by the relation

$$u_i = \frac{q_i}{\phi} \quad \text{where } i = x, y, z .$$

The terms q and ϕ are the volumetric flow rate and the effective porosity, respectively. The volumetric flow rate is the rate of volume flow or, the volume of fluid passing through a given area with time. Recall that the effective porosity differs from the intrinsic porosity: the intrinsic porosity is the total volume pore space within a medium, as the effective porosity is the estimate of the interconnected pore space in which fluid can travel.

By comparison, a *streamline* is a curve which is at every point tangent to the direction of the velocity vector of the fluid flow, at any *instant in time*. Mathematically speaking, a streamline is defined as a family of curves given by

$$\mathbf{u}(x, y, z, t_0) \times d\mathbf{S}(x, y, z) = 0$$

where the velocity is defined at a specific instant in time, t_0 , and $d\mathbf{S}$ is then an element of a curve along a streamline where x , y , and z are in the principal directions of the permeability. Streamlines may also be equivalently described by postulating an equipotential surface which will be presented in Section 2.2. Generally speaking, a streamline can be produced from any vector field.

Secondly, there is the concept of a *pathline*. A pathline is a line that connects the physical spatial path a particle would take as it traverses the medium, through the fluid. In steady flow, where the flow rates are established and invariant with time, pathlines and streamlines coincide and never intersect. However, with any change in the flow, they begin to separate: the streamlines are entirely altered with changing flow, as they are defined by the tangents of the velocity field at all points at an instant,

as the pathlines will simply alter route thereon, with the pathline being a physical representation that a particle would travel in space.

Thirdly, there is the concept of a *streakline*. A streakline represents the locations of a particle as it passes through a particular fluid spatial point within the medium with time. This is most easily pictured as a dye stream in a flowing fluid or as smoke released in the wind. The streakline would represent the physical location with respect to the moving fluid as the pathline represents the physical location in the stationary space. The streakline is a common term in the fluid dynamics and aerodynamics. Again, in steady flow, streaklines coincide with streamlines and pathlines, but with variance in flow, they all deviate and all three become distinct¹. Figure 1.1 illustrates streamlines, streaklines, and pathlines for an unsteady flow.

1.3 Streamline simulation in near-well

Streamline simulation for the purposes of reservoir engineering is of interest for a number of good reasons. Of utmost interest, being able to determine streamlines provide a visualization of the flow of fluid through the reservoir. In the case where you may have several, interconnected wells tapping into the same reservoir, this level of visualization, would easily identify the volume flow rates between injector and producer wells and, overall, obtain a better understanding of well communication.

Another key attribute of streamline simulation is its computational efficiency. In typical reservoir simulators, the flow equations require full three-dimensional finite difference solutions which tend to be computationally taxing. The fluid transport in streamline simulations reduce the problem to a one-dimensional system of equations along the streamlines, which are more easily solvable by numerical methods

¹The interested reader is pointed to [Bear, 1972], Section 6.5.1 for a rigorous description of streamlines, streaklines, and pathlines.

[King & Datta-Gupta, 1998].

Traditionally, streamlines are regarded as being well-suited for convective flow problems as opposed to diffusive flow where fluids don't have a well-defined flow direction. Streamlines reference the velocity field, so in cases where this velocity field isn't well-defined, streamlines are not particularly useful. Also, if the fluid flow contains strong transverse effects from the main direction of flow - for example gravity effects, which are dictated by density gradients - more elaborate mathematical methods need to be employed. However, standard finite difference methods are able to capture these effects quite well.

In the near-well region we are generally concerned with a very small region as compared to the reservoir. We are mostly concerned with fluid production or injection and visualization of the local well-completion and geological barriers. For this case, streamline simulation should be a prime tool with many possibilities.

1.4 Scope of this thesis

To improve the depiction of flow in the near-well region, a cylindrical grid system with a logarithmic radial-spacing of grid cells is utilized, oriented such that smaller radial steps are taken in nearer regions. Employing a cylindrical system takes advantage of the "natural" flow of the physical system, where the fluid flow converges at the wellbore as fluid is extracted.

Commonly, the wellbore information of the reservoir is encompassed within a single grid-block in the simulation model. In many cases, the well-block attempts to account for the influence of a well 10 centimeters in diameter, with a grid-block of square-lateral dimensions on the order of tens of meters, even in excess of 50 meters in some cases. Attempting to capture flow on the order of centimeters with a grid

dimension twice the order of magnitude will smear any notion of the complex flow geometries that exist in this region of the reservoir.

The near-well region of a drilled well requires particular attention as it quickly becomes critically complex. As examples, consider:

- while drilling takes place, drilling mud forces debris into the reservoir formation which impedes flow; the debris produces a 'damaged zone', where great efforts are employed in an attempt to account for the 'skin factor' value that disrupts flow into the well.
- upon insertion and cementing of the steel well-casing, a liner typically used to uphold the structure of a wellbore, perforations are sometimes blown through the casing and cement to allow fluids to flow. To produce such a perforation, controlled explosive propellants are used, and as such, the detonation sends a high-velocity shock through the liner and pushes metal, cement, and rock fragments into the surrounding formation. This creates a low-permeability "crushed zone" within the rock structure around the perforations by reducing the pore throats in this region and hindering flow.
- complex well completions - with the introduction of packers and wire-screens to withhold the production of sand, debris, etc. - aid in fluid production, however, make reservoir simulation in this region difficult and hard to predict. As the fluid flows toward the wellbore, frictional losses are associated with having to navigate a system of boundaries placed to regulate the flow. This region of the reservoir is typically void of detailed flow information, but may be a critical region as all produced flow must traverse this region, which is not necessarily the case for the remainder of the reservoir.
- the geological formation itself can be quite complex and there may be local flow

barriers in the near-well region that restrict flow from certain trajectories from the reservoir making for complex flow paths that are uncapturable during simulation employing typical computer models. This may be on top of perforations, infusion of drilling muds, etc.

None of these scenarios are capturable in typical reservoir simulators and would generally be estimated for their effects as a 'skin factor' coefficient to account for some of these details, and then only with history matching post-production would they be updated.

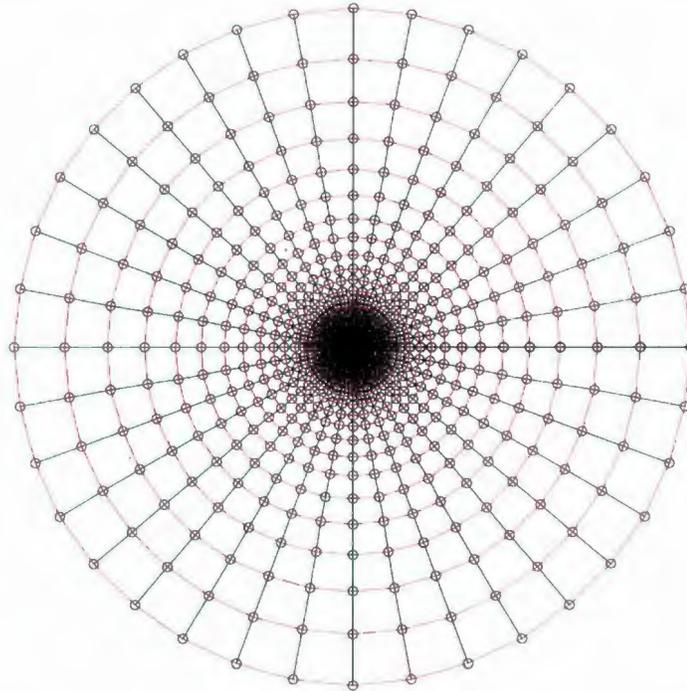


Figure 1.2: Areal view of the cylindrical grid structure to be overlaid co-axially with a wellbore model designed to simulate the flow trajectories of the near-well region. Streamline techniques will be defined to illustrate the flow of the constituent fluid as it navigates a simulated structure.

To model the near-well region, a radial - more precisely, polar cylindrical - grid model is aligned co-axially with the well trajectory where the radial dimensions of suc-

cessive grid-blocks will increase according to geometrical spreading, such that smaller radial steps are taken in the near-well and larger radial steps as we move outward. At this point, only constant angular steps are used to maintain computational simplicity during the simulations.

Herein, *streamline simulation* techniques are employed in a co-axial cylindrical grid as demonstrated in Figure 1.2. Streamline techniques are gaining interest in many industries due to its inherent efficiencies over purely numerical approaches. It has been extended into complex flow systems - e.g. compressible flow, inclusion of gravitational and capillary effects, miscible fluids for multi-phase flows, etc. - and shown to be able to compute flow on large grids even on many 'home computers'². The efficiency of streamline simulation is a result of its relatively low memory and computational requirements. Typical reservoir simulators require solving the entire grid-flux of all grid-blocks at once. In streamline simulators, only a single streamline based on the decoupled, single-dimensional transport equation is held in memory. The most widely accepted method of streamline computation is based on a semi-analytical approach developed by *Pollock* [Pollock, 1988]; the details of its usage will be discussed in detail in Section 2.3.2. Herein, an adaptation of the methodology devised by Pollock will be developed and applied to the geometry discussed above.

This work will expand the usage of streamline simulation, such that it is to be employed utilizing a cylindrical grid, with logarithmic radial cell spacings, and to be implemented for the near-well region within a reservoir. Prior to this study, these extensions are not discussed in literature.

²The extension of streamline simulations to include these complexities will be addressed in Section 2.4 on page 32.

Chapter 2

Mathematical Model

For our purposes, as this model is intended to be computationally efficient, it is only necessary to introduce an incompressible, single-phase reservoir model. This model provides all the necessary equations to define pressure distribution in the reservoir and is commonly implemented for many early-stage flow studies. Single-phase models are generally used to identify flow directions; identify connections between producers and injectors; in flow-based upscaling; in history matching; and in preliminary model studies. Methods of extending the single-phase model to include other relevant theory will be discussed in the Section 2.4.

2.1 Incompressible single-phase flow

Herein, we will construct the basic equations necessary to describe our medium. In light of keeping things simple, we will use the example of an incompressible reservoir – both medium and constituent fluid – and only single-phase homogeneous flow to keep computational lag to a minimum and describe the pressure distribution to determine the flow. To describe our medium, we need only begin with two equations, the *continuity equation* and *Darcy's equation* of flow.

The continuity is a mass balance relation connecting basic medium parameters and is given as

$$\frac{\partial}{\partial t} (\phi\rho) + \nabla \cdot (\rho\mathbf{u}) = q, \quad (2.1)$$

where ϕ and ρ are the effective porosity and fluid density, respectively, \mathbf{u} is the volumetric flux [length per time] and q volume flow rate which models the in/out flow [volume per time]. For our purposes, the system is incompressible and, in turn, the porosity and density terms are time independent. Hence, we may reduce equation (2.1) to

$$\nabla \cdot \mathbf{u} = \frac{q}{\rho}, \quad (2.2)$$

where we have rearranged the equation so as to isolate the divergence term.

In porous media flow, Darcy's equation relates an average volumetric flux - often referred to as the Darcy velocity or interstitial velocity - of a constituent fluid through the porous medium, and two of the prominent driving forces¹. Darcy's equation is given by

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu} (\nabla P + \rho g \nabla \mathbf{z}). \quad (2.3)$$

In Darcy's equation, there are two driving forces the gradient of pressure, ∇P , and a hydrostatic gravity term, $\rho g \nabla \mathbf{z}$. The gravity term, consisting of the acceleration due to gravity and the gradient of the vertical displacement, may be reduced to $-\rho g \Delta z$ in a conventional coordinate system to account for the hydrostatic pressure. Equation (2.3) illustrates that fluid travels from a region of high pressure to a region of lower pressure signified by the negative gradient, and scaled by the \mathbf{K}/μ term, where μ is the dynamic fluid viscosity and \mathbf{K} the intrinsic permeability. The permeability, \mathbf{K} , is a very important element in porous media which will be discussed in the following

¹We note here, that there may also be other forces at work on a constituent fluid in porous media - say, for example, capillary action - but, we will ignore such action heretofore. The interested reader is referred to [Bear, 1972], [Saatdjian, 2000], [Zheng & Bennett, 2002].

section.

Combining equations (2.2) and (2.3) yields the relation

$$\nabla \cdot \left[-\frac{\mathbf{K}}{\mu} (\nabla P - \rho g \nabla z) \right] = \frac{Q}{\rho}. \quad (2.4)$$

In the current framework, only horizontal fluid flow is considered as this is the general direction of flow within a reservoir, hence, equation (2.4) can be reduced further. Ignoring gravity and rearranging equation (2.4) yields

$$\nabla \cdot \mathbf{K} \nabla P = -\frac{Q\mu}{\rho}. \quad (2.5)$$

Up to this point, the intrinsic permeability and the fluid viscosity are taken to be constants and, therefore, are free to move them outside the differentiation. In having an incompressible medium - which includes the constituent fluid and rock matrix - we note that the viscosity and densities are constants.

In general, the permeability, \mathbf{K} , is a diagonally anisotropic tensor; meaning, the permeability in each direction may be dependent on the permeability in other directions. It is commonly accepted to reduce the intrinsic permeability into its principal components. Consequently, for reservoir simulation, the permeability tensor is diagonalized such that only the value of the permeability in the principal directions remain, namely, K_1 , K_2 , and K_3 for a three-dimensional coordinate system. In many cases this may be done without any loss of generality in the fluid flow². In practice, local variations of the permeability are implemented by introducing heterogeneities in the permeability matrix upon discretization.

It is sometimes useful to express the fluid properties in terms of the *hydraulic con-*

²Interested readers in the reduction of symmetric tensors to diagonal form are referred to [Arfken, 1985], Sec 3.4 and 4.6 (Diagonalization of Matrices).

ductivity, κ : a convenient expression used to combine media properties [Bear, 1972]. The hydraulic conductivity is a term that encompasses information regarding the rock matrix structure as well as the fluid information. It is defined, for an isotropic homogeneous fluid, as

$$\frac{\mathbf{K}}{\mu} = \frac{\kappa}{\rho g}, \quad (2.6)$$

where, ρ , and μ - the density, and viscosity terms, respectively - are taken to be constants for an incompressible homogeneous fluid³. We should recognize that the permeability is a tensor of rank two which ultimately defines the hydraulic conductivity as a tensor of rank two, which may be expressed in matrix notation as

$$\kappa = \begin{bmatrix} \kappa_{11} & \kappa_{12} & \kappa_{13} \\ \kappa_{12} & \kappa_{22} & \kappa_{23} \\ \kappa_{13} & \kappa_{23} & \kappa_{33} \end{bmatrix}.$$

Inspection of the variables in the definition of κ in equation (2.6), we see that the units of the conductivity are length per time - usually given in terms of meters per year (m/yr). Common values of hydraulic conductivity for water-flow range from $1 \times 10^7 m/yr$ for unconsolidated gravel to $1 \times 10^{-7} m/yr$ for certain shales and unfractured metamorphic rocks [Freeze & Cherry, 1979]. Substituting the conductivity into Darcy's equation (2.3) and rearranging we find

$$\begin{aligned} \mathbf{u} &= -\frac{\kappa}{\rho g} [\nabla P + \rho g \nabla z] \\ &= -\kappa \nabla \left[\frac{P}{\rho g} + z \right]. \end{aligned}$$

³The reader should note that the hydraulic conductivity definition in literature is often missing the gravity term g . Herein, the gravity is included for rigor, but throughout the document, gravity is ignored to concentrate our focus on the influence of pressure as the driving force. We are mainly concerned in horizontal fluid displacements, whereby Bernoulli's principle, fluid pressures are equal at equal depth. Interested readers are referred to [Zheng & Bennett, 2002, pp. 14-19].

Investigating this result, the term $P/\rho g$ represents the pressure per unit weight of fluid. This is equivalent to the net work done by a unit weight of incompressible fluid against the pressure difference along its flow. The z term is the height above a datum level. The sum of these two terms define the *piezometric head* of the medium,

$$\phi = \frac{P}{\rho g} + z .$$

This is a common term used in permeability and hydraulic conductivity testing⁴. Combining the expressions of Darcy's law and piezometric head yields

$$\mathbf{u} = -\kappa \nabla \phi , \tag{2.7}$$

which is a standard expression for potential flow.

The hydraulic conductivity tensor is, as well, a symmetric tensor, meaning that it has only *six* independent components; e.g. $\kappa_{12} = \kappa_{21}$, $\kappa_{13} = \kappa_{31}$, and $\kappa_{23} = \kappa_{32}$ [Bear, 1972].

Furthermore, the hydraulic conductivity is commonly reduced to its principal components - aligned in the principal directions. Using the method of eigenvalue decomposition allows the reduction of any symmetric tensor to a diagonal one; cases where the tensor to be diagonalized is non-symmetric, singular value decomposition may be employed to do so [Zijl, 1996].

It may be shown, without any loss of generality, that the conductivity tensor, κ , may be diagonalized to principal components, usually aligned along the coordinate axes. Given that the hydraulic conductivity tensor is symmetric, it is mathematical fact that it may be diagonalized for any orthogonal coordinate system [Morse &

⁴The piezometric head ϕ , is used when piezometer data is employed. A piezometer is a tube perforated at the end which is inserted into the medium to measure the fluid rise against gravity, and measure static pore-pressure.

Feshbach, 1953, pp. 21-31]. In literature, the permeability and conductivity are generally given in terms of the principal components [Aziz & Settari, 1979, pp. 12].

In the case of an isotropic medium, the conductivity, \mathbf{K} , being simply a scalar constant is not affected by differentiation. So, in view of equation (2.5), the expression may be reduced to the following:

$$\nabla^2 P = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2} = -\frac{Q}{\mathbf{K}}. \quad (2.8)$$

For the case of anisotropic conductivity – in that the conductivity is reduced to its principal components, $\mathbf{K} = \mathbf{K}(K_1, K_2, K_3)$, equation (2.5) becomes

$$K_1 \frac{\partial^2 P}{\partial x^2} + K_2 \frac{\partial^2 P}{\partial y^2} + K_3 \frac{\partial^2 P}{\partial z^2} = -Q. \quad (2.9)$$

2.2 Streamfunctions and streamline tracing

In previous chapters, the notion of flow through a porous medium is purely thought of as flow forced by imposed pressure⁵ with Darcy's equation wholly describing the direction of flow; noting that Darcy's law relates the average volumetric flux in a porous medium to the gradient of pressure. The concept of fluid flow in porous media, defined by Darcy's law, is an Eulerian approach of describing flow of a fluid through porous media. The notion of streamfunctions, streamlines, and particle tracking, are Lagrangian concepts based on the trajectory of the locus of particles within the fluid as it travels through the medium.

⁵We note that pressure is the most prominent fluid driving force, however, gravity and capillary action also play a key role in an actual physical system. Herein, for brevity and simplicity, these have been neglected.

2.2.1 Classical streamfunctions

The streamfunction is a concept that is, in certain respects, the reverse of other methods that have been described thus far. Essentially, fluid flow in porous media is based on the existence of a pressure gradient that causes flow; motion of the fluid is described by Darcy's equation which relates a fluid flux to the gradient of pressure. Conversely, we may also employ the streamlines to determine the fluid flux through the medium⁶.

In many classical formulations, the streamfunction is defined via complex analysis; here we will introduce the streamfunction in a more concise method. To do so, an introduction to the streamfunction is done rigorously in a two-dimensional system, with only a brief extension to illustrate the equivalency in three-dimensions. This will allow us to better relate it to the physical system and understand its ramifications to the flow regime.

2.2.1.1 Isotropic streamfunction in two dimensions

The most intuitive way of describing the streamfunction is to first consider a very simple system, two-dimensional, steady state flow of an incompressible fluid within a homogeneous and isotropic medium. Under Darcy flow of a homogeneous fluid, the system is satisfied by Laplace's equation, given by

$$\nabla^2 P = \frac{\partial^2 P(x, y)}{\partial x^2} + \frac{\partial^2 P(x, y)}{\partial y^2} = 0, \quad (2.10)$$

⁶It is important to recognize here that the fluid flux in a porous medium expressed by Darcy's law is to be viewed as an average concept, as Darcy's law is defined by an empirical law; although, Darcy's law has been verified by homogenizing the Navier-Stokes equation and assuming stationarity, creep, and incompressible flow [Bear, 1972, pp. 173].

where we have set the source term to zero for simplicity. Given the linearity of differential operators, equation (2.10) may also be equivalently expressed as

$$\frac{\partial}{\partial x} \left(\frac{\partial P(x, y)}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial P(x, y)}{\partial y} \right) = 0. \quad (2.11)$$

By inspection of the form of equation (2.11), we are able to satisfy this expression by introducing the following function:

$$\frac{\partial P(x, y)}{\partial x} = \frac{\partial \psi(x, y)}{\partial y}, \quad (2.12)$$

$$\frac{\partial P(x, y)}{\partial y} = -\frac{\partial \psi(x, y)}{\partial x}. \quad (2.13)$$

where $\psi(x, y)$ denotes the streamfunction. We may note that equations (2.12) and (2.13) satisfy equation (2.11) by the equality of mixed partials assuming sufficient smoothness.

Considering the Darcy velocities in the x - and y -directions, as previously discussed, we have the following expressions for the average fluid velocities:

$$u_x = \left(-\frac{K_x}{\mu} \right) \frac{\partial P(x, y)}{\partial x},$$

$$u_y = \left(-\frac{K_y}{\mu} \right) \frac{\partial P(x, y)}{\partial y}.$$

To define a streamline, we can say that we wish to define a line which is tangent to the fluid direction at all spatial points; hence, the slope of the streamline must be equivalent to the ratio of the orthogonal fluid velocities. We may employ the Darcy velocities to describe the local slope,

$$\frac{dy}{dx} = \frac{u_y}{u_x} = \frac{\partial P(x, y) / \partial y}{\partial P(x, y) / \partial x} = -\frac{\partial \psi(x, y) / \partial x}{\partial \psi(x, y) / \partial y}. \quad (2.14)$$

Equivalently, equation (2.14) may be rewritten as

$$\left(\frac{\partial\psi(x,y)}{\partial x}\right) dx + \left(\frac{\partial\psi(x,y)}{\partial y}\right) dy = 0. \quad (2.15)$$

From equation (2.15) we can immediately recognize that this is the form of the chain rule for the total differential of $\psi = \psi(x, y)$, explicitly,

$$d\psi(x, y) = \left(\frac{\partial\psi(x, y)}{\partial x}\right) dx + \left(\frac{\partial\psi(x, y)}{\partial y}\right) dy.$$

Henceforth, we can conclude that from these conditions that

$$d\psi(x, y) = 0, \quad (2.16)$$

and accordingly, the streamfunction, $\psi(x, y)$, must be constant along a streamline, e.g.

$$\psi(x, y) = \text{constant} .$$

A question now arises: If $\psi(x, y)$ is constant, how are $\psi(x, y)$ and $P(x, y)$ related? From vector analysis, given any function, say $f(x, y) = 0$, taking the gradient of $f(x, y)$ produces a function orthogonal to the level set curve; which, in a two-dimensional system, $\nabla f(x, y)$ is orthogonal to $f(x, y)$ and would point in the direction of greatest ascent. If we take the gradients of the streamfunction, $\psi(x, y)$, and pressure, $P(x, y)$, we may test their orthogonality by taking their dot product. From vector analysis, the dot product of two orthogonal vectors should vanish to zero. By taking the dot product the gradients,

$$\nabla\psi(x, y) \cdot \nabla P(x, y) = \left[\frac{\partial\psi(x, y)}{\partial x}, \frac{\partial\psi(x, y)}{\partial y}\right] \cdot \left[\frac{\partial P(x, y)}{\partial x}, \frac{\partial P(x, y)}{\partial y}\right],$$

and substituting equations (2.12) and (2.13) we find that

$$\left[\frac{\partial \psi(x, y)}{\partial x}, \frac{\partial \psi(x, y)}{\partial y} \right] \cdot \left[\frac{\partial \psi(x, y)}{\partial y}, -\frac{\partial \psi(x, y)}{\partial x} \right] = 0.$$

This is a critical result illustrating the relation between $\psi(x, y)$ and $P(x, y)$ for an isotropic medium, this indicates that the two gradients are mutually orthogonal. Meaning, the streamfunction, $\psi(x, y)$, and the pressure, $P(x, y)$, are mutually orthogonal in a homogeneous and isotropic medium under steady state conditions. To emphasize the importance of the streamfunction a little further, does $\psi(x, y)$ satisfy some differential equation of flow itself? In taking the derivative of equation (2.12) with respect to x and equation (2.13) with respect to y , we have

$$\frac{\partial}{\partial y} \left(\frac{\partial P(x, y)}{\partial x} = \frac{\partial \psi(x, y)}{\partial y} \right) \Rightarrow \frac{\partial^2 P(x, y)}{\partial x \partial y} = \frac{\partial^2 \psi(x, y)}{\partial y^2},$$

$$\frac{\partial}{\partial x} \left(\frac{\partial P(x, y)}{\partial y} = -\frac{\partial \psi(x, y)}{\partial x} \right) \Rightarrow \frac{\partial^2 P(x, y)}{\partial y \partial x} = -\frac{\partial^2 \psi(x, y)}{\partial x^2}.$$

By equality of mixed partials, we can therefore conclude

$$\frac{\partial^2 \psi(x, y)}{\partial x^2} + \frac{\partial^2 \psi(x, y)}{\partial y^2} = \nabla^2 \psi(x, y) = 0$$

namely, $\psi(x, y)$ satisfies Laplace's equation. Hence, the streamfunction may be used equivocally with pressure to describe the same system of fluid flow.

A fundamental property of the streamfunction is that $\psi(x, y)$ remains constant along the streamline. The streamfunction itself is an analytical function, meaning it may be exploited to produce analytical streamline paths. However, to do so requires the fluid-flow be divergence-free. In reservoir simulation, because the source term (q) is zero everywhere except at injection and production points, this is an acceptable

condition, as we can break the region of simulation into homogeneous pieces and analytically trace only those sections not containing the injection/production sites.

2.2.1.2 Anisotropic streamfunction in two dimensions

As an extension of the previous section, the description of steady state flow for isotropic permeability, we will introduce an anisotropic permeability.

As stated in Section (1.1.2) the steady state flow for an anisotropic flow in two-dimensions is described by the equation

$$K_x \frac{\partial^2 P(x, y)}{\partial x^2} + K_y \frac{\partial^2 P(x, y)}{\partial y^2} = 0,$$

or equivalently

$$\frac{\partial}{\partial x} \left(K_x \frac{\partial P(x, y)}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y \frac{\partial P(x, y)}{\partial y} \right) = 0. \quad (2.17)$$

Applying the same methodology as for the isotropic case, we note that we may satisfy equation (2.17) by introducing the streamfunction as

$$K_x \frac{\partial P(x, y)}{\partial x} = \frac{\partial \psi(x, y)}{\partial y},$$

$$K_y \frac{\partial P(x, y)}{\partial y} = -\frac{\partial \psi(x, y)}{\partial x}.$$

Introducing the Eulerian Darcy velocities as in the previous section,

$$u_x = - \left(\frac{K_x}{\mu} \right) \frac{\partial P(x, y)}{\partial x} = - \left(\frac{1}{\mu} \right) \frac{\partial \psi(x, y)}{\partial y}$$

and

$$u_y = - \left(\frac{K_y}{\mu} \right) \frac{\partial P(x, y)}{\partial y} = \left(\frac{1}{\mu} \right) \frac{\partial \psi(x, y)}{\partial x},$$

we define the local slope as

$$\frac{dy}{dx} = \frac{u_y}{u_x} = -\frac{\partial\psi(x, y)/\partial x}{\partial\psi(x, y)/\partial y} \quad (2.18)$$

Rewriting equation (2.18), we may express it as

$$\left(\frac{\partial\psi(x, y)}{\partial x}\right) dx + \left(\frac{\partial\psi(x, y)}{\partial y}\right) dy = 0.$$

As before, using the total differential of $\psi = \psi(x, y)$ leads to the implication that

$$d\psi(x, y) = 0,$$

the streamfunction is constant along a streamline for the anisotropic permeability case as well as the isotropic previous discussed.

There is a very noticeable difference in defining and analyzing the isotropic and anisotropic streamfunction. Testing for orthogonality, demonstrates the difference quite clearly. Then, as before, by taking the dot product of the gradients of the pressure and streamfunction for the anisotropic case, we can determine if the two are orthogonal, explicitly,

$$\begin{aligned} \nabla\psi(x, y) \cdot \nabla P(x, y) &= \left[\frac{\partial\psi(x, y)}{\partial x}, \frac{\partial\psi(x, y)}{\partial y}\right] \cdot \left[\frac{\partial P(x, y)}{\partial x}, \frac{\partial P(x, y)}{\partial y}\right] \\ &= \left[\frac{\partial\psi(x, y)}{\partial x}, \frac{\partial\psi(x, y)}{\partial y}\right] \cdot \left[\frac{1}{K_x} \frac{\partial\psi(x, y)}{\partial y}, -\frac{1}{K_y} \frac{\partial\psi(x, y)}{\partial x}\right] \\ &= \frac{\partial^2\psi(x, y)}{\partial x\partial y} \left[\frac{K_y - K_x}{K_x K_y}\right]. \end{aligned} \quad (2.19)$$

Since, in equation (2.19), the dot product does not vanish to zero, $\psi(x, y)$ and $P(x, y)$ are not orthogonal. We instead find that there is a remaining mixed-partial term. It is worth noting that the mixed partial exists purely due to the anisotropy, whereby

the $(K_y - K_x)/K_x K_y$ coefficient and reduces to zero in the case where $K_x = K_y$.

2.2.2 Streamfunctions in three-dimensions

It is important to consider the streamfunction in three dimensions and illustrate that streamline theory, in two-dimensions, projects easily into the three-dimensional world.

To describe a flow in three-dimensions in terms of the streamfunction requires defining two functions,

$$\psi(x, y, z) = \text{const}, \quad \text{and} \quad \chi(x, y, z) = \text{const} .$$

On their respective contours, the bi-stream functions, ψ and χ , form two-dimensional *stream-surfaces* in which the streamlines identically exist along the lines of intersection of the stream-surfaces. As previously noted, the streamlines are instantaneously tangent to the fluid velocity at each point.

It can be shown that the the bi-stream functions are related the to average Darcy velocity as

$$\rho \mathbf{u} = \nabla \psi \times \nabla \chi \tag{2.20}$$

The stream-surfaces $\psi(x, y, z)$ and $\chi(x, y, z)$ are actually the characteristic curves of the equation

$$\mathbf{u} \times d\mathbf{r} = 0$$

where $d\mathbf{r}$ is an element of the arc along a streamline.

As illustrated in Figure 2.1, the stream-surfaces bound a region of flow within the medium whereby fluid travels tangentially along the streamlines and never cross the surfaces. As previously discussed, the two-dimensional case with flow in the xy -plane, the plane $z = \text{constant}$ played the role of the second stream surface. It can

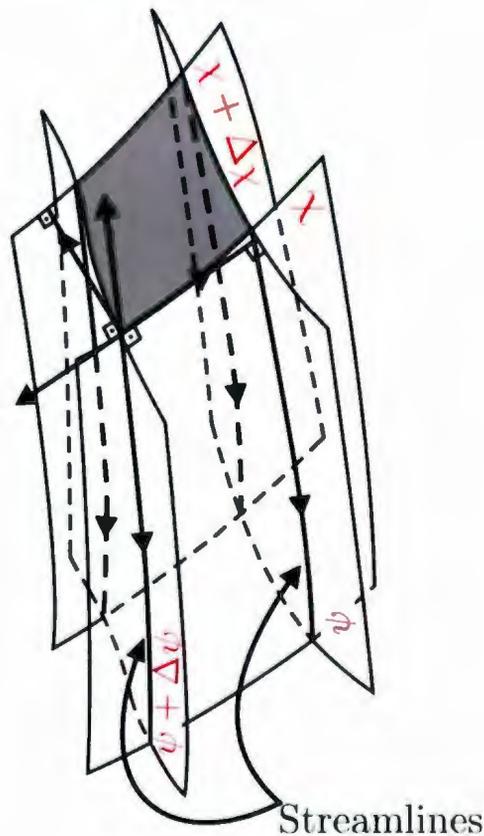


Figure 2.1: The stream-surfaces $\psi(x, y, z)$ and $\chi(x, y, z)$ intersect at the streamlines, which are tangent to the averaged fluid-velocity vector. The cross-sectional area bounded by the stream-surfaces - shaded above - are mass-conservative [Bear, 1972].

be shown that the extension to three-dimensions is governed by the same rules as that of the two-dimensional case [Bear, 1972, Datta-Gupta & King, 2007]. From this construct, the *streamtube* that is created by the bounding stream functions, allows for an analytical determination of the volume flux, due to the fact that this system is mass conservative. This method requires a multitude of boundary conditions and is only determinable analytically in highly simplified flow regimes. In turn, numerical methods must be employed to solve this system.

2.3 Streamline tracing and particle tracking techniques

In much of streamline literature, especially related to reservoir simulation, the time-of-flight method (TOF) is the most predominant method of streamline generation. This method is based on finding the travel-time required, denoted herein as τ , for a particle to travel a distance, s , along a streamline.

At a fundamental level, the TOF derivation is based on the transformation of the physical coordinate system into one which follows the fluid flow direction; explicitly, the flow direction is defined by the bi-stream functions ψ and χ and the flight coordinate τ [Bear, 1972, King & Datta-Gupta, 1998]. Due to this result and knowing that \mathbf{u} is mutually orthogonal to $\nabla\psi$ and $\nabla\chi$, we come to the result

$$\mathbf{u} \cdot \nabla_{\psi,\chi,\tau} = \phi \frac{\partial}{\partial \tau}, \quad (2.21)$$

where $\mathbf{u} \cdot \nabla_{\psi,\chi,\tau}$ is the directional derivative of the velocity [Hægland, 2009]. Equation (2.21) may be expressed along the length of the streamline

$$\frac{\partial \tau}{\partial s} = \phi \frac{\mathbf{u}}{\|\mathbf{u}\|^2}.$$

The streamline path is based on the velocity field of the fluid within the medium. Explicitly, the streamline method is based on the following integral:

$$\tau(x, y, z) = \int_0^s \frac{\phi}{\|\mathbf{u}\|} ds, \quad (2.22)$$

where τ is the required time, ϕ the porosity, u the scalar velocity, and s the path

length along the streamline. Or, in differential form

$$\mathbf{u} \cdot \nabla \tau(x, y, z) = \phi \quad (2.23)$$

Knowing $\mathbf{u} = \nabla \psi + \nabla \chi$ and substituting this into equation (2.23) we find

$$(\nabla \psi + \nabla \chi) \cdot \nabla \tau = \phi ,$$

which is an expression for the Jacobian for a three dimensional change of variables [Arfken, 1985].

In equation (2.22), $\tau(x, y, z)$ is the time in which a particle initially at zero would travel a distance s along the streamline. Physically, we have defined a new coordinate system, (ψ, χ, τ) , whereby the streamlines are now straight lines measured in units of τ , where the pore volume is conserved and reduces the computation of the three-dimensional flow equations into a sum of single-dimensional transport equations [King & Datta-Gupta, 1998, Datta-Gupta & King, 2007].

2.3.1 Streamline tracing

The critical point in the simulation, will be the details and development of the cylindrical streamline tracing techniques. An adaptation of a method presented by Pollock is used to implement the actual tracing of the streamline [Pollock, 1988].

It should be noted that Pollock's method is not, strictly speaking, a streamline trace, but rather a particle tracking method along a pathline. To distinguish between the streamline and pathline formulations, recall their definitions from Chapter 1. A streamline is defined as the line connecting the tangents of a velocity vector field at an instant in time, where in unsteady flow, would require to be updated as the flow

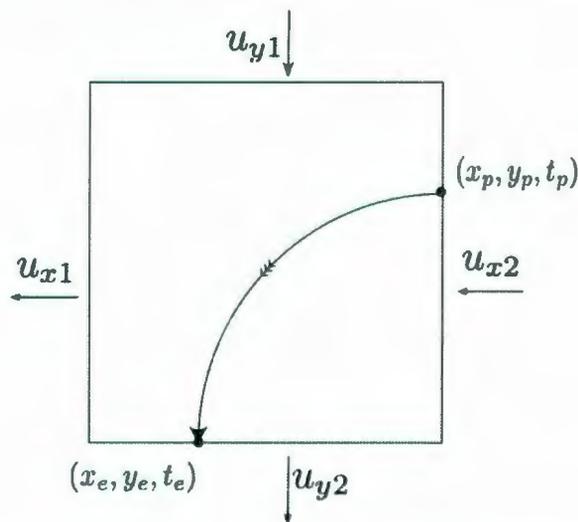


Figure 2.2: Illustration demonstrating the Pollock methodology for a single Cartesian block. Here the face velocities for each side is show by u_{x1} , u_{x2} , u_{y1} , and u_{y2} . As well, the pathline and entry/exit points through the block are shown.

changes. Whereas, a pathline follows the physical path of a particle within the flow where its direction vector identifies the velocity vector at that point at some previous moment in time.

2.3.2 Pollock's method in 2D Cartesian

Pollock's method for the tracking ground water flow is one of the most widely accepted methodologies for streamline generation in porous media [Pollock, 1988]. The method is robust and presented as a semi-analytical approach to the tracing of a pathline through porous a medium. As in the classical approach to stream function tracking, Pollock's method hinges on assuming a velocity profile, calculating the volumetric flux across the face of an unit-square grid block, using these to determine the travel-time through the block, and estimate an imaginary particle's exit point.

For the case of steady-state flow, the flow characteristics are invariant with time, and both the streamlines and pathlines coincide; for these circumstances streamline tracing is equivalent to particle tracking in flows. For our purposes here, we will

consider a two-dimensional system, as the extension to three-dimensions is trivial.

Consider a unit square as in Figure 2.2. Given that the flow is steady, the face velocities u_{x_1} , u_{x_2} , u_{y_1} , and u_{y_2} are also invariant with time. The particle enters the unit square at point (x_p, y_p) at time t_p on an arbitrary edge - this is the initial position of the particle within the cell. If we assume that the average velocity change through the square is linear in each direction, it is sufficient to assume that the velocity profile through the block can be expressed as

$$u(x) = A_x(x(t) - x_p) + u_{x_1} \quad (2.24)$$

$$u(y) = A_y(y(t) - y_p) + u_{y_1} \quad (2.25)$$

for the x - and y -components of the velocity. Writing the differential expression of equation (2.24) and applying the chain rule we have

$$\frac{du(x)}{dt} = \frac{du(x)}{dx} \frac{dx}{dt} .$$

Noting that dx/dt is the definition of velocity in the x -direction and differentiating equation (2.24) directly with respect to x , we come to the expression

$$\left(\frac{du(x)}{dt} \right)_p = A_x u_{xp} .$$

This means that the average velocity of the particle p at any time can be defined by the constant A_x and the velocity in the x -component u_{xp} . Rearranging and integrating yields the result

$$\begin{aligned} \int_{t_1}^{t_2} A_x dt &= \int_{u_{x_1}}^{u_{x_2}} \left(\frac{1}{u_{xp}} \right) du(x) \\ A_x \Delta t_x &= \ln [u_{x_1}/u_{x_2}] . \end{aligned}$$

Solving this equation for Δt yields an expression for the travel-time of a particle to traverse the block in the x -direction. Similarly, the same expression can be found for the y -direction, yielding

$$\Delta t_x = (1/A_x) \ln[u_{x2}/u_{x1}] , \quad (2.26)$$

$$\Delta t_y = (1/A_y) \ln[u_{y2}/u_{y1}] . \quad (2.27)$$

Rearranging equation (2.27) and substituting equations (2.24) to solve for $x(t)$ and $y(t)$, we have

$$x_e = x_p + (1/A_x) (u_{xp} \exp[A_x \Delta t_x] - u_{x1}) , \quad (2.28)$$

$$y_e = y_p + (1/A_y) (u_{yp} \exp[A_y \Delta t_y] - u_{y1}) . \quad (2.29)$$

These expressions now give us an analytical methodology of finding an exit location (x, y) for where the imaginary particle will exit the block given an entrance location, as well as the time it will take to traverse the block in each direction.

When calculating the travel-time for equations (2.27), two values are yielded. Similarly to Fermat's principle in optics, the path of least time indicates the exit face; if $t_x < t_y$, then the particle would exit in the x -direction. Once the travel-times are calculated from the face velocities, the least time value is then used to calculate the displacement the particle will travel in each corresponding direction.

This methodology is quite powerful in its construction as it is quite robust. As each exit point for a block is calculated, this point can then be carried forward as the entrance point for the adjacent block which shares that interface, with each exit point being carried forward as the new entry point until a production point is reached within the grid. With the collection of points, a particle pathline can be visualized.

However, it is not without issues. There is one oversight that can be quite troublesome: the particle cannot exit a block on the same face as it entered. The most obvious solution to the problem would be to refine our grid to smaller cells such that the streamline is forced to encounter more grid interfaces to track its trajectory. This increases the computational time which can, but not necessarily, nullify one of the primary advantages of this method. As with all reservoir simulation techniques, care must be taken to balance the amount of information that can be achieved with the computational capability required to return the information in a timely manner.

2.4 Literature review

The current framework of petroleum streamline simulation roots in fluid-modeling by authors such as Muskat and Wyckoff [Muskat & Wyckoff, 1934]. They proposed a tracing technique for a “line-drive” path of fluid based on a mathematical model for a steady pressure distribution for a water-flooded reservoir. Ultimately, hoping to determine the connection between injection and production wells.

By the 1960s, analytical streamfunctions derived for homogeneous, steady-state flow in porous media were well described. With this, the fluid-flux components through a medium could then be expressed in terms of the streamfunction [Zaslavsky, 1962]. A rigorous derivation of the classical streamfunction in porous media for a homogeneous fluid was readily available [Bear, 1972] and the notion of using streamline methods in reservoir simulation was quite promising and sought-after.

From those early beginnings, much work has been done to bring streamline modeling to its current state. Of particular importance was the work of Pollock. This approach to pathline determination is based on a mass-balance relation that is determined from finite difference methods to calculate fluid-flux through a given area on

a Cartesian grid. Conservative fluxes are obtained at the grid faces from a discrete pressure calculation. Having the fluxes of a closed cell and then employing a linear velocity interpolation scheme, a continuous velocity profile is developed through the block. With that, an analytical integration technique is introduced such that, given any entrance point to the cell, a travel-time, referred to as the time of flight, to each potential exit face within a cell can be determined. Imposing the condition that the least time calculation will determine the exit face, the displacement of the “particle” in each direction given by the travel-time [Pollock, 1988].

Pollock’s method is based on a simplified method for single-phase, steady-state flow of an incompressible fluid through a porous medium. It is designed for a square Cartesian grid, ignoring gravity and diffusion effects. Its methodology is extremely robust and effective and is the inspiration of most reservoir streamline models today; though Pollock’s method has been much expanded by various authors to help overcome some of its short-fallings.

During the 1990s, the use of streamline simulation techniques became of particular interest to the petroleum reservoir industry as a means of gaining more insight into the fluid-flow in the subsurface. With geological/geophysical data advancing so rapidly, typical reservoir simulation grids were becoming quite large and cumbersome, and a more computationally efficient methodology was sought. As such, a multitude of literature on the advancements of streamline modeling exists and contain many major expansions of this method. The interested reader is directed to [Datta-Gupta & King, 2007], as it includes a detailed description of many of the advancements made in in streamline simulation.

Multi-phase flow is one of the most useful extensions of the theory as streamline methods are often employed for water- or polymer-flooding of reservoirs for hydrocarbon displacement. In multi-phase flow, we have simultaneous flow of more than

one fluid through the medium. To include the presence of more than one fluid, new equations need to be included in the mathematical model. On top of the continuity and Darcy's equations, we must introduce the concept of saturation content of each fluid. With this, we assume that the medium is always saturated by some mixture of reservoir fluids (either miscible or immiscible). The saturation term S is,

$$\sum S_f = 1$$

where f represents each phase of fluid within the medium. With this, there are several levels of detail that need to be introduced. The continuity equations are then expanded to include each fluid and the fraction of the fluid located at each point, e.g. the fractional flow. The continuity equation then becomes

$$\frac{\partial S_f}{\partial t} + \mathbf{u}_f \cdot \nabla f_f = 0, \quad (2.30)$$

where $\mathbf{u} = \sum \mathbf{u}_f$ is the total fluid velocity, and f_f is the fractional flow of each phase of fluid. Essentially, it is assumed that each fluid exists at all locations within the medium and the presence of each fluid is described by an equation of the mass fraction at a point; the sum of the mass fractions of all fluids within the medium are 1 [Aarnes *et al*, 2007, Bratvedt *et al*, 1993, Bradvedt *et al*, 1996, Batycky *et al*, 1996].

It is important to notice that introducing multi-phase flow into the flow generation, increases the number of equations - as there must be a specific mass-balance for each fluid - but the overall solution to the problem does not increase computationally. As illustrated in Section 2.2, the equation (2.30) can be changed into the time-of-flight coordinates whereby

$$\mathbf{u} \cdot \nabla = \phi \frac{\partial}{\partial \tau}$$

where τ is the time of flight. Combining this with equation (2.30) and we find that the saturation equation is reduced to

$$\frac{\partial S_f}{\partial t} + \frac{\partial f_f}{\partial \tau} = 0$$

which has reduced the three-dimensional fluid flow into a series of single-dimensional equations in τ for the saturation S_f along a streamline. This result is the basis of multi-phase flow regimes for streamline simulation due to its general analytical form and computational stability.

A few contributing authors have made tremendous leaps into bringing streamline simulation into the limelight in the petroleum industry. Of note is the work of Bratvedt *et al.* which extended work into a full 3D streamline simulation for a two-phase flow which included a gravity correction [Bratvedt *et al.*, 1996]. This method was groundbreaking in that it was the first field-scale full simulation of its kind to include gravity. The basis of the model presented was to solve for the streamline based on the saturation expression and apply a “gravity correction” step based on the convective transport of the fluids for the the time-step knowing the phase densities within a medium. The reasoning for this method is that when gravity is included in the theory, the system becomes 2D - one direction is defined along the streamlines and the other along the gravity. An operator splitting technique is employed to numerically separate the two dimensions by solving for the saturation excluding gravity, and to use this solution for a given time step to determine the effect of the gravity during that time. This method was also employed and independently verified by Batycky *et al.* for a full field-scale simulation to be used for a commercially available reservoir suite [Batycky *et al.*, 1996].

The extension to compressible flow is typically a complicated process in reservoir

simulation, as solving the flow equations in three dimensions for a time dependent porosity and density increase computational times considerably. However, Cheng *et al.* present a rigorous treatment of the streamline for multi-phase flow [Cheng *et al.*, 2005]. In compressible streamline flow, it remains true that the streamlines are tangent to the velocity vectors at all points and defined by the expression

$$\rho_{eff}\mathbf{u} = \nabla\psi + \nabla\chi , \quad (2.31)$$

where ρ_{eff} is referred to as an effective density as it may vary with changing physical conditions. This is now a total mass-flux conservative expression whereby, for incompressible flow $\rho_{eff} = 1$ and the system reduces to the standard model. In view of the vector identity

$$\nabla \cdot (\nabla\psi + \nabla\chi) = 0 ,$$

we can then substitute (2.31) and from the product rule find

$$\phi \frac{\partial \rho_{eff}}{\partial \tau} + \rho_{eff} \nabla \cdot \mathbf{u} = 0 ,$$

which now allows the definition of the ρ_{eff} along a streamline given any initial value at a reference location [Cheng *et al.*, 2005]. With this expression, the streamline may then be integrated along each cell since there is no requirement that $\nabla \cdot \mathbf{u} = 0$, and instead equal to the sum of the total volume flux. Given a velocity interpolation scheme, this method can remain as a semi-analytical method of introducing compressibility to any streamline simulation.

A very significant portion of streamline simulation and particle tracking methods numerical by definition. As all of these expressions need to be ported and discretized into programming languages, all equations must be written as difference equations

for discrete data. In doing so, the selection of the grid in which these equations are discretized, introduces an alternate side to this problem.

Here, we have chosen a cylindrical grid with logarithmically spaced radial boundaries such that the grid blocks remain as uniform as possible with geometric spreading. However, streamline simulation has also been extended into irregular grid structures. [Hægland *et al*, 2006] developed a methodology in which Pollock's method is extended to curvilinear coordinates by using a mapping algorithm - an isoparametric trilinear transformation - to map each grid cell in physical space into a unit cube in the reference space. The theory is then tested and compared to several discretization techniques and the results compared to reduce the numerical dispersion introduced in highly irregular grid geometries.

Streamline techniques have also been adapted to be effective in unsteady flow conditions. In this case, the streamlines are no longer constants, as the pressure distribution alters, so to does the trajectory of the streamlines themselves. In which case, each streamline will then require periodic updating to compensate for the changing pressure field. This work is well described by Batycky [Batycky *et al*, 1996].

In this work, a simplified physical model is chosen and focus is instead on the near wellbore region of the reservoir. Careful attention is put on deriving the physical equations for an anisotropic permeability field structured on a co-axial, radial grid overlaid on the wellbore. As this region has not previously been studied using these techniques, this system is intentionally kept simplified to study the result of the concept, with the intention of adding more structural information in future research.

Chapter 3

Streamline Tracing in Cylindrical Coordinates

In the forthcoming chapter, we introduce an adaptation of Pollock's particle tracing technique specifically designed for simulating the flow for the near-well region utilizing a cylindrical coordinate system. The design of the method has been chosen in such a way as to take advantage of the natural geometry of a wellbore; to depict the wellbore, we chose a radial-cylindrical model such that the axis of the simulation grid is overlaid co-axially with the well, allowing the native geometry of the cylindrical coordinate system to encompass the flow regime in this region.

Drawing from current methods of streamline simulation a *new* cylindrical streamline model is presented that is specifically formulated for the near-well region of a wellbore. The results described in the following chapter, was recently published, with applications to some more detailed well-completion models [Skinner & Johansen, 2011].

3.1 Cylindrical transform

To begin the transformation to the cylindrical coordinate system, we must first transform the flux equation from the standard Cartesian to cylindrical coordinates.

We begin with the Darcy's equation, namely,

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu} \nabla P ,$$

where \mathbf{u} is the Darcy velocity, \mathbf{K} is the intrinsic permeability of the medium, and μ the dynamic fluid viscosity, as usual. Explicitly, the Darcy equation may be written in matrix form as

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = -\frac{1}{\mu} \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial x} \\ \frac{\partial P}{\partial y} \\ \frac{\partial P}{\partial z} \end{bmatrix} . \quad (3.1)$$

We now must transform the Cartesian expression of Darcy's law to the cylindrical coordinate system. The cylindrical coordinate system is described by the transformation

$$\begin{aligned} x = r \cos \theta & \quad \leftrightarrow \quad r^2 = x^2 + y^2 \\ y = r \sin \theta & \quad \leftrightarrow \quad \theta = \arctan \left(\frac{y}{x} \right) \\ z = z & . \end{aligned}$$

To apply this transformation to equation (3.1) into cylindrical, we must first determine the expression of the gradient in cylindrical coordinates. Knowing that the gradient is defined as

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

Using the chain rule, the partials for x and y may be rewritten as

$$\begin{aligned}\frac{\partial}{\partial x} &= \frac{\partial}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial}{\partial \theta} \frac{\partial \theta}{\partial x} = \cos \theta \frac{\partial}{\partial r} - \frac{\sin \theta}{r} \frac{\partial}{\partial \theta}, \\ \frac{\partial}{\partial y} &= \frac{\partial}{\partial r} \frac{\partial r}{\partial y} + \frac{\partial}{\partial \theta} \frac{\partial \theta}{\partial y} = \sin \theta \frac{\partial}{\partial r} + \frac{\cos \theta}{r} \frac{\partial}{\partial \theta},\end{aligned}$$

and the z term remains invariant. Hence, the gradient in cylindrical coordinates becomes

$$\nabla_{r\theta z} = \left(\cos \theta \frac{\partial}{\partial r} - \frac{\sin \theta}{r} \frac{\partial}{\partial \theta}, \sin \theta \frac{\partial}{\partial r} + \frac{\cos \theta}{r} \frac{\partial}{\partial \theta}, \frac{\partial}{\partial z} \right). \quad (3.2)$$

We see here that the components of the gradient here include both radial and angular components. We would like to isolate these to only the r and θ directions, exclusively. To do so, we will require the unit vectors, namely, $\hat{\mathbf{r}} = \mathbf{r}/\|\mathbf{r}\|$ and $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}/\|\boldsymbol{\theta}\|$. From vector calculus, these are given as

$$\begin{aligned}\hat{\mathbf{r}} &= [\cos \theta, \sin \theta, 0], \\ \hat{\boldsymbol{\theta}} &= [-\sin \theta, \cos \theta, 0], \\ \hat{\mathbf{z}} &= [0, 0, 1].\end{aligned}$$

Using the unit vectors and the dot product, we find that the gradient in the $\hat{\mathbf{r}}$ and $\hat{\boldsymbol{\theta}}$ directions are

$$\begin{aligned}\frac{\partial}{\partial r} &= \hat{\mathbf{r}} \cdot \nabla_{r\theta z} = \frac{\partial}{\partial r}, \\ \frac{\partial}{\partial \theta} &= \hat{\boldsymbol{\theta}} \cdot \nabla_{r\theta z} = \frac{1}{r} \frac{\partial}{\partial \theta}.\end{aligned}$$

As $\hat{\mathbf{z}}$ is simply $[0, 0, 1]$, given that there is no change in this direction.

Now to determine the direction of the velocity in the $\hat{\mathbf{r}}$ direction, we similarly use

the dot product by

$$u_r = \hat{\mathbf{r}} \cdot \mathbf{u} = [\cos \theta, \sin \theta, 0] \cdot -\frac{1}{\mu} \left[K_x \frac{\partial P}{\partial x}, K_y \frac{\partial P}{\partial y}, K_z \frac{\partial P}{\partial z} \right],$$

substituting expression (3.2), and doing some algebra, we find that

$$u_r = -\frac{1}{\mu} \left[(K_x \cos^2 \theta + K_y \sin^2 \theta) \frac{\partial P}{\partial r} + (K_y - K_x) \sin \theta \cos \theta \frac{1}{r} \frac{\partial P}{\partial \theta} \right]. \quad (3.3)$$

Similarly, the velocity in the θ direction can be determined by

$$u_\theta = \hat{\boldsymbol{\theta}} \cdot \mathbf{u} = [-\sin \theta, \cos \theta, 0] \cdot -\frac{1}{\mu} \left[K_x \frac{\partial P}{\partial x}, K_y \frac{\partial P}{\partial y}, K_z \frac{\partial P}{\partial z} \right],$$

which yields the result

$$u_\theta = -\frac{1}{\mu} \left[(K_y - K_x) \sin \theta \cos \theta \frac{\partial P}{\partial r} + (K_x \sin^2 \theta + K_y \cos^2 \theta) \frac{1}{r} \frac{\partial P}{\partial \theta} \right]. \quad (3.4)$$

From equations (3.3) and (3.4) and seeing that $u_z = \hat{\mathbf{z}} \cdot \mathbf{u} = K_z \partial P / \partial z$, we can now express Darcy's equation in matrix form as follows

$$\begin{bmatrix} u_r \\ u_\theta \\ u_z \end{bmatrix} = -\frac{1}{\mu} \begin{bmatrix} K_x \cos^2 \theta + K_y \sin^2 \theta & (K_y - K_x) \sin \theta \cos \theta & 0 \\ (K_y - K_x) \sin \theta \cos \theta & (K_x \sin^2 \theta + K_y \cos^2 \theta) & 0 \\ 0 & 0 & K_z \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial r} \\ 1/r \frac{\partial P}{\partial \theta} \\ \frac{\partial P}{\partial z} \end{bmatrix}. \quad (3.5)$$

For brevity of equation (3.5), we will set

$$K_r = K_x \cos^2 \theta + K_y \sin^2 \theta, \quad (3.6)$$

$$K_t = (K_y - K_x) \sin \theta \cos \theta, \quad (3.7)$$

$$K_\theta = K_x \sin^2 \theta + K_y \cos^2 \theta, \quad (3.8)$$

where K_r can now be referred to as the radial permeability, K_θ the angular permeability, and K_t the tangential permeability as it is a tangential component to the velocity direction.

Equation (3.5) yields a connection between the permeability tensors from the Cartesian to the cylindrical coordinate system; as in Cartesian coordinates, the permeability has only principal components, the transformation to cylindrical has off-diagonal values. On closer investigation of the off-diagonal values of the permeability, take for example for the radial flux, namely,

$$u_r = (K_x \cos^2 \theta + K_y \sin^2 \theta) \frac{\partial P}{\partial r} + [(K_y - K_x) \sin \theta \cos \theta / r] \frac{\partial P}{\partial \theta} ,$$

the bracketed terms illustrate the contribution of the permeability, which was defined in x - and y -directions, but herein translated to the radial coordinate system. Taking $0 \leq \theta \leq \pi/2$: the coefficient $K_x \cos^2 \theta + K_y \sin^2 \theta$ has a greater contribution in K_x for $0 \leq \theta \leq \pi/4$, and conversely, from $\pi/4 \leq \theta \leq \pi/2$, the contribution of K_y is greater. However, upon transformation of the radial flux equation from Cartesian, we notice that u_r now has an angular component – noted by the coefficient of $\partial P / \partial \theta$ is now non-zero – given by $K_t = (K_y - K_x) \sin \theta \cos \theta$. Due to the trigonometric dependency, K_t will have a maximum contribution of $1/2(K_y - K_x)$ at an angle of $\pi/4$ and its iterants in each quadrant, namely, $3\pi/4$, $5\pi/4$, and $7\pi/4$.

If, for example, we have an isotropic system - e.g. $K_x = K_y = K_z$ - we note that the permeability matrix in equation (3.5) is reduced to a very simple expression. The off-diagonal terms of the permeability become zeros and the diagonal terms simplify.

Explicitly,

$$\begin{bmatrix} u_r \\ u_\theta \\ u_z \end{bmatrix} = -\frac{1}{\mu} \begin{bmatrix} K & 0 & 0 \\ 0 & K & 0 \\ 0 & 0 & K \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial r} \\ 1/r \frac{\partial P}{\partial \theta} \\ \frac{\partial P}{\partial z} \end{bmatrix},$$

as $K = K_x = K_y = K_z$ and taking advantage of the trigonometric identity $\sin^2 \theta + \cos^2 \theta = 1$. Hence, for an isotropic system, the cylindrical transformed equation does not increase in complexity compared to that of the Cartesian one.

3.2 Cylindrical velocity interpolation scheme

Given the expression for Darcy's equation in cylindrical coordinates, equation (3.5), we have a velocity value for the volumetric flux for this geometry. We now wish to investigate the flow of fluid through the medium for the cylindrical coordinate system. It will be useful to understand the connection between the velocity and pressure. To employ a streamline tracing technique for this geometry, we will need to choose an appropriate velocity interpolation scheme.

3.2.1 Radial velocity interpolation

Considering steady-state flow in only the radial direction, Darcy's equation may be expressed as

$$u_r = \frac{q}{A} = -\frac{K_r}{\mu} \frac{\partial P}{\partial r}. \quad (3.9)$$

In one-dimensional flow of an isotropic system, the volumetric flow rate q is distributed along the circumference of a circle where the expression then becomes

$$q = -2\pi r \frac{K_r}{\mu} \frac{dP}{dr}.$$

Integrating, we find that the pressure as a function of radius may be expressed as

$$P(r) = \left(-\frac{q\mu}{2\pi K_r} \right) \ln(r) + \text{const} . \quad (3.10)$$

Here we see that the pressure is proportional to the natural log of the radius, explicitly

$$P(r) \propto A \ln(r) + Br .$$

Knowing that $u_r \propto \partial P / \partial r$, we can then say that the volumetric flux profile would be adequately described by

$$u(r) \propto \frac{A}{r} + B .$$

In light of this, the interpolation we will use for the radial velocity component is

$$u(r) = \frac{A_r}{r} + \beta , \quad (3.11)$$

where

$$A_r = r_1 r_2 \frac{(u_{r2} - u_{r1})}{(r_1 - r_2)} \quad \text{and} \quad \beta = \frac{A_r}{r_1} + u_{r1} .$$

The subscripts for the radius and velocity values indicate the boundary values of their respective parameter; e.g. a radial boundary at a radius r_1 , will have an associated velocity denoted u_{r1} , and so on.

Equation (3.11) will be used to determine the flow velocity at any radius within each grid block, and the Darcy velocity, equation (3.9), will define the velocity value at grid boundaries. In doing so, we will have a velocity field that is quasi-continuous throughout the simulation area which allows us to semi-analytically interpolate the flow through each block.

3.2.2 Angular velocity interpolation

For the angular component, the pressure does not directly depend on the angle for such a system as in equation (3.10), and hence, would be constant at all angles for a given radius for an isotropic system. So, for a velocity interpolation scheme in the angular direction, we could assume that the pressure variation would be reasonably smooth and therefore choose a linear average for this model. Following the work of [Pollock, 1988], we can employ a simple linear interpolation scheme to produce a continuous velocity distribution in the angular direction given by

$$u(\theta) = A_\theta(\theta - \theta_1) + u_{\theta_1} \quad (3.12)$$

where

$$A_\theta = \frac{u_{\theta_2} - u_{\theta_1}}{\Delta\theta} .$$

Equations (3.11) and (3.12) will be the velocity interpolation profile for the streamline flowing through a grid-block in the radial and angular directions, respectively.

Using the interpolation expressions (3.11) and (3.12), for the radial and angular interpolations, respectively, our goal now is to use these expressions to calculate a “time-of-flight” of a neutral particle through the medium. The goal is to determine the exit point given the velocities and time. Given an expression for the velocity through some medium, integrating yields an expression for its position in time. Hence, we must integrate the respective velocity interpolations in order to solve for position.

3.2.3 TOF and radial displacement

From equation (3.11), we have a velocity interpolation scheme for the radial flow given boundary velocities and radii, namely,

$$u(r) = -\frac{A_r}{r} + \beta . \quad (3.13)$$

This expression indicates that the velocity interpolation profile of the flow in the radial direction will be assumed to follow a logarithmic change in pressure. As fluid moves radially, say from outer to inner boundaries, due to geometric convergence, velocity is additive; the increase of velocity is inversely proportional to the radius from the wellbore.

Recognizing that the velocity $u(r)$ is the rate of change of the radius with time, e.g.

$$u(r) = \frac{dr(t)}{dt} ,$$

we may solve for the time based on the radial displacement and velocity. So, by integrating expression (3.13) in time,

$$\int_{t_1}^{t_2} dt = \int_{r(t_1)}^{r(t_2)} \frac{1}{u(r)} dr(t) = \int_{r(t_1)}^{r(t_2)} \left(\frac{1}{A_r/r(t) + \beta} \right) dr(t) ,$$

$$\Delta t = \frac{1}{\beta} \left[\int_{r(t_1)}^{r(t_2)} \frac{A_r + \beta r(t)}{A_r + \beta r(t)} dr(t) - \int_{r(t_1)}^{r(t_2)} \frac{A_r}{A_r + \beta r(t)} dr(t) \right] .$$

Carrying through the integration then yields

$$\Delta t_r = t_2 - t_1 = \frac{1}{\beta} [r(t_2) - r(t_1)] + \frac{A_r}{\beta^2} \left(\ln \left[\frac{\beta r(t_2) - A_r}{\beta r(t_1) - A_r} \right] \right) \quad (3.14)$$

where, as before,

$$A_r = r_1 r_2 \frac{(u_{r2} - u_{r1})}{(r_2 - r_1)} \quad \text{and} \quad \beta = \frac{A_r}{r_1} + u_{r1} .$$

Equation (3.14) is an expression for the “time-of-flight” through a radial section, bounded by r_1 and r_2 , such that we may approximate a neutral particle’s path based on the fluid velocity values at the boundaries, namely u_{r1} and u_{r2} .

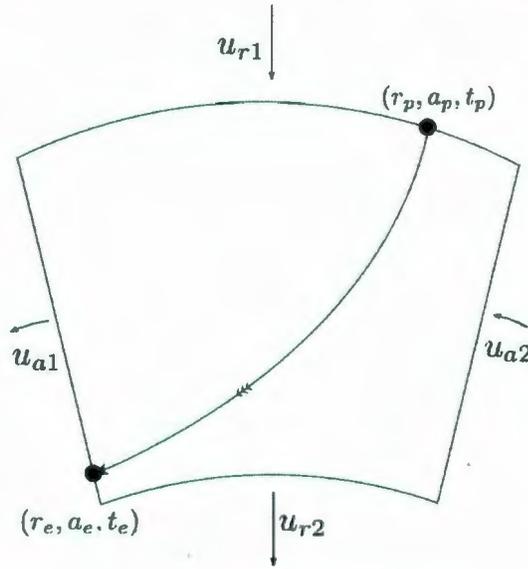


Figure 3.1: Radial annular region. Here we have a illustration of an annular region of the cylindrical grid-block. We note the radial and angular velocity values and arrows associated with an arbitrary flow direction; if flow is found to not be in the direction chosen, the associated values will be negative. An imaginary particle, in the case shown here, enters at the block at the point (r_p, θ_p, t_p) signifying a radius r_p , an angle θ_p , at a time value t_p . As the particle passes through the block, determined based on the face velocities, u_{r1} , u_{r2} , $u_{\theta1}$, and $u_{\theta2}$, exits the block at (r_e, θ_e) at time t_e .

Considering Pollock’s method, the exit location was determined from the time-of-flight by simple algebra. In view of equation (3.14), we see that in attempting to solve for $r(t_2)$, no immediate solution is applicable¹. In light of this, however, we will

¹A solution of the time of flight Δt_r is possible using the Lambert- W function, otherwise known as the Omega function. It is an inverse function of $f(w) = we^w$ where e is the exponential function and w is a complex number. It however, cannot be expressed using elementary functions and hence,

instead apply a more “physical” approach.

We are well aware of the equations of kinematics to describe motion in constant acceleration, namely,

$$r(t) = r_0 + \int_0^t \mathbf{v} dt' = r_0 + \int_0^t (v_0 + \mathbf{a}t') dt' .$$

If we assume that the acceleration in the radial direction is roughly constant through the radial grid-block, then we could employ this equation. Knowing the initial position and the initial velocity on that face of the block we can set

$$r_0 = r_p \quad \text{and} \quad v_0 = u(r_p) ,$$

and that the particle would travel a time Δt_r our equation becomes

$$r(t) = r_p + u(r_p)\Delta t_r + \int_0^{\Delta t_r} a_r t' dt' .$$

Now, using the definition of acceleration and the chain rule, we may approximate the acceleration as

$$a_r = \frac{du(r)}{dt} = \frac{\partial u(r)}{\partial r} \frac{dr}{dt} = \frac{\partial u(r)}{\partial r} u(r) .$$

Since our velocity interpolation in the radial component is $u(r) = A_r/r + \beta$, differentiating this with respect to r , we can then say

$$a_r = -\frac{A_r}{r^2} u(r) .$$

is of little use in the following section where a discretization scheme will be developed.

Hence, the exit radius r_e will then become

$$r_e = r_p + u(r_p)\Delta t_r - \frac{1}{2} \left(\frac{A_r}{r^2} u(r) \right) \Delta t_r^2, \quad (3.15)$$

where the value for the radius r and the velocity $u(r)$ will be taken to be an average value taken from the radial center of the block. More on this in Chapter 4.4.

3.2.4 TOF and the angular displacement

For the angular velocity interpolation, it is sufficient to have a linear average as in Pollock's methodology [Pollock, 1988] by which the angular velocity expression

$$u(\theta) = A_\theta(\theta(t) - \theta_1) + u_{\theta 1} \quad (3.16)$$

where

$$A_\theta = \frac{u_{\theta 2} - u_{\theta 1}}{\Delta\theta}$$

We recognize that in a flow scenario, a particle's angular position will be a function of its time-of-flight, e.g. $\theta = \theta(t)$.

The change in u_θ with time for a neutral particle may be expressed as

$$\left(\frac{du_\theta(t)}{dt} \right)_p = \left(\frac{du_\theta}{d\theta} \right) \left(\frac{d\theta(t)}{dt} \right)_p \quad (3.17)$$

where the subscript p indicates the specific particle. In a linear interpolation, the change of the velocity u_θ with angular position, may be represented by the angular difference between the beginning and end points; in such a case, we may write

$$\left(\frac{du_\theta}{d\theta} \right) = \frac{u_{\theta 2} - u_{\theta 1}}{\Delta\theta} = A_\theta .$$

From expression (3.17) we note that we have the derivative of the angular position θ with respect to time for a particle at that position. Given that we have an imaginary particle traveling through an annular region, we may note that by definition

$$u_{\theta p} = \frac{d\theta(t)}{dt} ,$$

where the $u_{\theta p}$ denotes the angular component of the velocity for the particle at the angular location $\theta(t)$. Hence, we may re-write expression (3.17) as

$$\left(\frac{du_{\theta}(t)}{dt} \right)_p = A_{\theta} u_{\theta p} . \quad (3.18)$$

Rearranging equation (3.18) and integrating from $t_1 \leq t \leq t_2$ we find

$$\int_{t_1}^{t_2} \frac{1}{u_{\theta p}} du_{\theta}(t) = \int_{t_1}^{t_2} A_{\theta} dt .$$

Since, A_{θ} is independent of t , it may be taken outside the integration, and as such the result becomes

$$A_{\theta} \Delta t_{\theta} = \ln \left[\frac{u_{\theta p}(t_2)}{u_{\theta p}(t_1)} \right] . \quad (3.19)$$

Herein, Δt_{θ} is representative of the amount of time required for an imaginary particle to traverse an angle θ for a grid block, given the interfacial velocity values and the linear interpolation scheme in equation (3.16). So, we now have an expression for the time-of-flight through an annular region of the cylindrical system given by the expression (3.19); solving, we have

$$\Delta t_{\theta} = \frac{1}{A_{\theta}} \ln \left[\frac{u_{\theta p}(t_2)}{u_{\theta p}(t_1)} \right] . \quad (3.20)$$

Substituting equation (3.16), namely $u_{\theta p}(t_2) = A_{\theta} [\theta_p(t_2) - \theta_1] + u_{\theta 1}$, into (3.20) above,

taking the exponential, and solving for $\theta_p(t_2)$ yields the result

$$\theta_p = \frac{u_\theta(t_1)}{A_\theta} [\exp(A_\theta \Delta t_\theta) - 1] + \theta_1 . \quad (3.21)$$

The velocity values $u_{\theta p}(t_1)$ and $u_{\theta p}(t_2)$ are the cell face velocities at the boundaries θ_1 and θ_2 which are known based on the flux calculations taken from the pressure distribution of the field.

3.3 Steady-state flow and the TOF

We should recall that we have assumed a steady-state flow regime. Meaning, the flow of fluid through the medium is invariant with time, which, in turn, means that an established flow will remain unchanged and all associated velocities remain as they are. Since the velocities are constant, then the streamlines are established based on the velocity field and do not require any further updating.

Under the assumption of steady-state flow, the integrated time-of-flight (TOF) calculations yield a travel-time for a neutral particle to traverse a simulated region. Recollecting what equations (3.14) and (3.20) physically represent help in understanding which of the time values are most meaningful and help determine some other information regarding the particle's path. For the interpolation, we assumed that the velocity profile for the radial and angular directions would follow an inverse proportionality in r and a linear average in the angular region θ , respectively; explicitly, $u_r \propto 1/r$ in the radial direction and $u_\theta \propto \theta$ in the angular. Having continuous velocity profiles in the radial and angular directions, we can logically arrive to the conclusion that the minimal time should indicate that the particle is traveling more quickly in that respective direction; say, if $\Delta t_r < \Delta t_\theta$, then an imaginary particle p initially at radius r_1 , will exit the face of the annular region through face r_2 . The times t_r and

t_θ , are solved based on the physical dimensions and face velocities of a closed cell.

Having the TOF tells us the travel-time for the particle through the cell, and this time value is then carried into equations (3.15) and (3.21) to determine the exit location of the particle in (r, θ) through the block. This position is then carried forward to the adjacent block where it becomes the new entrance point and new face velocities are calculated, and so forth, until a discharge point is met.

Chapter 4

Discretization For Implementation

Based on theory introduced in Chapter 3, a discretization scheme is presented to trace streamlines through rudimentary flow regimes.

As illustrated in Figure 4.1, we will only consider a single quadrant of the full areal pressure field. This reduces the actual computations necessary to a quarter of its original amount without any loss of generality of the simulation as the pressure distribution is symmetric about the principal axes. The pressure equation is solved on a large Cartesian grid and the appropriate pressure corresponding to the node locations are extracted from the full pressure field.

In Figure 4.1, a contour map of the pressure field for a full Cartesian grid is shown for an anisotropic permeability field with the pressure nodes of the cylindrical grid overlaid. We note that for a given angle, there are nodes for several radii and as the radius decreases toward the well, the node density increases significantly. These nodes correspond to the cell centers of the logarithmic radial grid. How these nodes are determined is discussed in Section 4.3.

Figure 4.1 *b*) is a contour map generated from the cylindrical grid points. We note here that the contours in both the Cartesian and the radial geometries of pressures

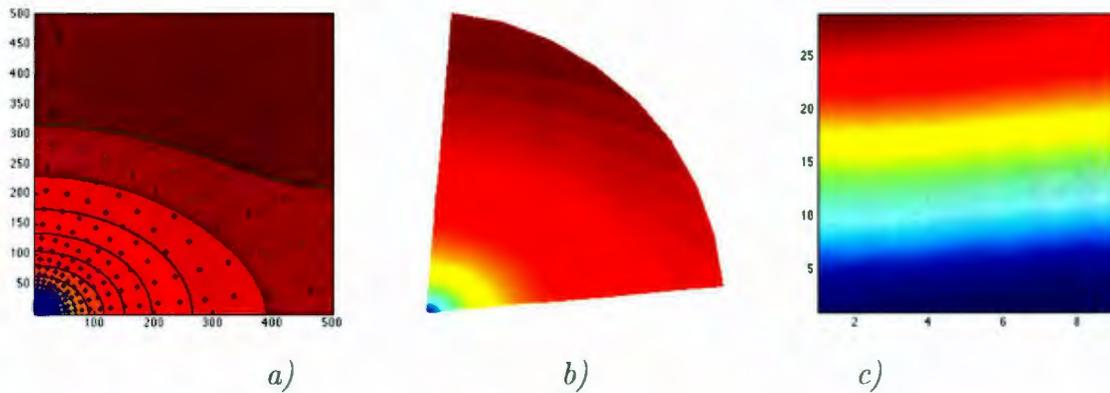


Figure 4.1: Pressure contours for simulated region. Red colour indicates relative high pressure, blue relative low. Plot *a)* shows the pressure contours for the first quadrant of a simulated anisotropic porous medium with a flow sink at block (1,1) with the node locations plotted. Plot *b)* are the pressure contours constructed using only the cylindrical node pressure values from the square Cartesian system. Plot *c)* is the node data plotted as a Cartesian array. The detailed code used to produce these plots is provided in Appendix A and B.

are quite similar and illustrate that the same pressure trends exist.

The final plot, on the right in Figure 4.1 *c)*, is a Cartesian contour plot based on the radial pressure node data from the previous plot. When plotted in Cartesian co-ordinates, we see that the contours become curved due to the anisotropy. This plot was mainly was used to illustrate that the geometry of the radial pressure data was complete.

In the forthcoming sections, a discretization of the theory is presented for a two-dimensional horizontal system that may be implemented in a simple routine on an “off-the-shelf” computer; particular attention will be paid to how such theory could be put into practice using Matlab’s programming language¹.

¹There are a multitude of programming languages that may be used to implement a routine such as this. The Matlab environment was chosen as much of the mathematical construct that is required may be called upon as built-in functions; the language of Matlab also tends to be readable to most persons with experience in computational sciences with minimal effort.

4.1 Discretization for pressure simulation

A simple discretization of equation (2.5) is presented for the purposes of simulating pressure in an isotropic and anisotropic medium as outlined by [Aarnes *et al*, 2007]. For simplicity and computational efficiency, a two-point flux approximation (TPFA) is used to determine the pressure distribution.

The TPFA method uses two points – the cell averages – to approximate the flux through the cell volumes². Herein, we will use a regular, square grid design aligned along the coordinate axes in the xy -plane. In doing so, we maintain a highly efficient implementation of the TPFA method.

Our system of equations are of the form

$$\nabla \cdot u = -\nabla \cdot \lambda \nabla P = \frac{q}{\rho}, \quad (4.1)$$

where $\lambda = \mathbf{K}/\mu$ (where we ignore gravity effects) is the fluid mobility and the term q accounts for an sources or sinks within the system. Our discretization will be designed to solve for P .

So, let's start by denoting the interface between two cells by γ_{ij} and pick the positive x -direction on which to begin, namely the normal $n_{ij} = (1, 0)^T$. With this, equation (4.1) becomes

$$u_{ij} = - \int_{\gamma_{ij}} (\nabla \cdot \lambda \nabla P) \cdot n_{ij} dV. \quad (4.2)$$

²Although here we are strictly concerned about the two-dimensional flow, for the purposes of calculating the flow-flux, an arbitrary cell thickness is chosen. So, although the pressure calculation is a 2D simulation, the cells do in fact have a volume.

We may approximate the derivative of P on γ_{ij} in the direction $n_{ij} = (1, 0)^T$ by

$$\nabla P \approx \delta P = \frac{2(P_j - P_i)}{\Delta x_i - \Delta x_j}. \quad (4.3)$$

We note here that Δx references the cell dimension in the x -direction and this calculation will have to be repeated in the y -direction.

For the discretization of the TPFA method, the permeability matrix, \mathbf{K} , within λ , is not well defined at the interfaces. Hence, to approximate, we will use a distance-weighted average of the directional cell permeabilities; where λ_i at interface γ_{ij} , is denoted by $\lambda_{i,ij}$. We have

$$\lambda_{i,ij} = n_{ij} \cdot \lambda_i n_{ij} \quad \text{and} \quad \lambda_{j,ij} = n_{ij} \cdot \lambda_j n_{ij}. \quad (4.4)$$

With expressions (4.4), we may define the permeability in the direction of n_{ij} of λ_{ij} on the interface γ_{ij} . For example, in the x -direction, the expression becomes

$$\lambda_{ij} = (\Delta x_i + \Delta x_j) \left(\frac{\Delta x_i}{\lambda_{i,ij}} + \frac{\Delta x_j}{\lambda_{j,ij}} \right)^{-1}. \quad (4.5)$$

Combining equations (4.3) and (4.4) with the velocity equation (4.2) we have

$$u_{ij} = -|\gamma_{ij}| \lambda_{ij} \delta P_{ij} = -2|\gamma_{ij}| \left(\frac{\Delta x_i}{\lambda_{i,ij}} + \frac{\Delta x_j}{\lambda_{j,ij}} \right)^{-1} (P_j - P_i). \quad (4.6)$$

Using expression (4.6), and repeating it in the y -direction, we are now able to employ Matlab for our simulation. For the detailed Matlab code see Appendix A.

4.2 Discretization Code in Matlab

Employing the TPFA method, we are able to determine the pressure values throughout a section of media in both isotropic and anisotropic permeability fields. The test set-up is as follows: a square grid system aligned with the coordinate axes was chosen; a single cell in the near-center region of the grid system will contain the sink to simulate the well; no-flow boundary conditions at outer boundaries.

Given that the TPFA method is one of the simplest finite volume methods there are, it has some notable drawbacks when compared to other more complex systems. In literature, numerical dispersion tends to be a large factor in the computational error introduced by this approximation and for the same reason it is so efficient, is its drawback – its simplicity. However, for pressure determination in steady-state flow, the TPFA is generally considered sufficient [Aarnes *et al*, 2007].

4.2.1 Isotropic medium

The isotropic Matlab code is shown in Figure 4.2. This code is used to set up the grid dimensions, indicated by *Grid.Nx* and *Grid.Ny* at the top of the code, and also assembles the correct dimensions for the permeability matrix using these values. Also, we use this code to define our source/sink in the grid. Here we notice that q is defined as being zero everywhere except at a cell near the center of the system, where q has been defined to be arbitrarily set to -1 . This illustrates a “sink” placed in the center where a flow is occurring.

The illustration in Figure 4.3 show two plots yielded by Matlab using the code given in Figure 4.2 for the isotropic case. For an isotropic medium, we expect there to be an excellent symmetry in the pressure distribution, in all directions within the plane, outward from the sink. The symmetry is a result of the permeability matrix

```

% To set the Grid size in each direction.
Grid.Nx = 501; Grid.lx = 1/Grid.Nx;
Grid.Ny = 501; Grid.hy = 1/Grid.Ny;
Grid.Nz = 1; Grid.hz = 1/Grid.Nz;

%To set the permeability grid size.
Grid.K = ones(3,Grid.Nx, Grid.Ny);

% our actual grid dimensions.
N = Grid.Nx* Grid.Ny* Grid.Nz;

% The flow is described by:
q = zeros(N,1);
q([(0.5*N + round(0.5*Grid.Nx))]) = [-1];

% The pressure is then invoked by the command
P = TPFA (Grid, Grid.K,q);

```

Figure 4.2: This is the Matlab code employed to set up the grid dimensions and flow conditions for the isotropic case. We note that the last line in the code calls upon the *TPFA2* function that we defined in the previous section.

being equal in all cells, and therefore, the flux over all interfaces should be equal. As Figure 4.3 shows, the pressure distribution is smooth and concentric about the sink at the center of the plot.

4.2.2 Anisotropic medium

In the anisotropic code, Figure 4.4, the grid is defined and the anisotropic permeability is introduced as K_x being 60% that of K_y and is assembled into a matrix. As in the isotropic code, the grid size and the permeability field is fed into the TPFA function and pressure is calculated based on the direction and the total flow q . We also note that the flow, q , is arbitrarily assigned a value of -1 , signifying the sink.

As to be expected, the pressure contours in Figure 4.5 are circular in the vicinity of the sink and elliptical as they approach the outer boundary. This is purely the

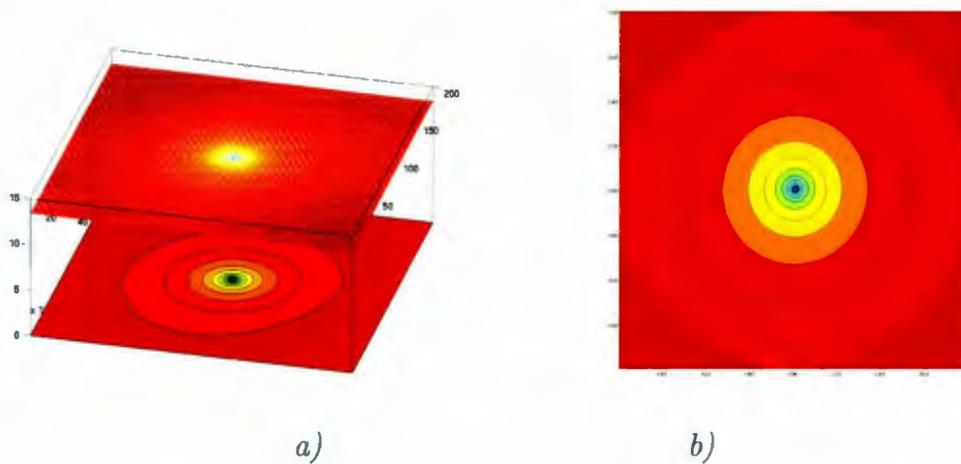


Figure 4.3: Result of having isotropic permeability on the pressure for TPFA simulation in Matlab with a sink placed near the center of the grid system. Plot *a)* illustrates the grid layout as the top layer and the projected pressure contours on the lower. Plot *b)* shows the isolated contour plot of the pressure distribution around the fluid sink mimicking the pressure drop due to a production well.

result of the anisotropy of the permeability field.

4.3 Radial grid discretization

At the center of this concept is the construction of a grid system for the approximation of the radial flow problem such that, a high level of detail is obtained close to the wellbore so as to capture minute changes in the flow as well as be able to carefully map geological properties in this region.

There are a multitude of errors that can be associated with grid size and orientation when the grid is being designed, however, of equal importance is grid regularity and size distribution of adjacent blocks. It is important to ensure that adjacent block size transitions are smooth and that care is taken to keep the grid blocks as regular as possible. Placing a large grid block adjacent to a relatively small grid block can create relatively large numerical errors [Logan, 2002].

Herein, a system was developed such that a starting radius is chosen nearest the

```

% To set the Grid size in each direction.
Grid.Nx = 200; Grid.hx = 1/Grid.Nx;
Grid.Ny = 200; Grid.hy = 1/Grid.Ny;
Grid.Nz = 1; Grid.hz = 1/Grid.Nz;
kx = 0.6;
ky = 1;

% To set the anisotropic permeability grid size.
K1 = ky .* ones(1,Grid.Nx);
K2 = kx .* ones(1,Grid.Ny);
Kt = ([K1; K2; ones(1,Grid.Nx)]);
Grid.K = Kt(:,ones(1,1,Grid.Nx));

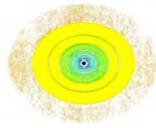
% our actual grid dimensions.
N = Grid.Nx* Grid.Ny* Grid.Nz;

% The flow is described by:
q = zeros(N,1);
q([(0.5*N + round(.5*Grid.Nx))]) = [-1];

% The pressure is then invoked by the command.
P = TPFA2 (Grid, Grid.K,q);

```

Figure 4.4: This is the Matlab code for the anisotropic case. Here, we note the assembling of the permeability matrix denoted *Grid.K* to define the anisotropy in the simulation.



system au

are

are

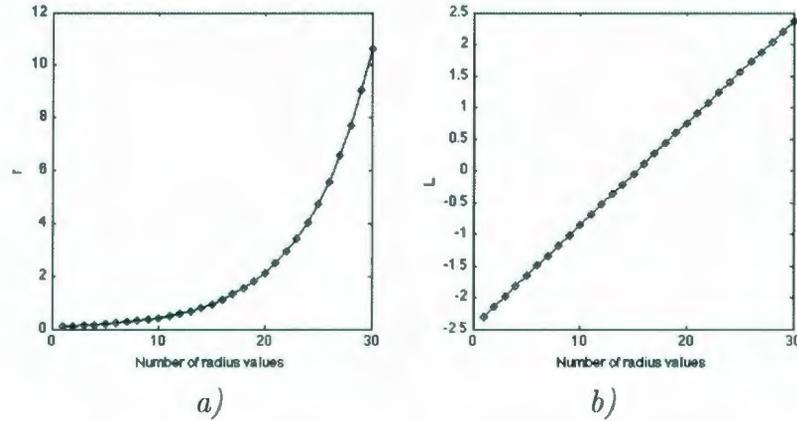


Figure 4.6: Plot *a)* is radius versus number of points illustrating the radius values produced by the algorithm $r_{n+1} = r_n + r_n \Delta\theta$ properly space the radial steps according to the chosen initial radius and angular width of the blocks. Plot *b)* shows that a linear plot is produced by plotting of the natural log of radius vs the number of points, illustrating that the radii values are logarithmic.

would like. Given that the radial pressure varies logarithmically, this should produce relatively regular steps in the pressures as we approach the wellbore.

Since, in this system, a two-point flux approximation (TPFA) is chosen, we will need to precisely determine the block nodes at the appropriate centers. To do so, we follow the work of [Aziz & Settari, 1979] and employ a logarithmic mean to define the radial node locations. The logarithmic mean radius for the node positions is

$$r_i = \frac{r_{n+1} - r_n}{\ln(r_{n+1}/r_n)} . \quad (4.8)$$

To now build a suitable grid structure, we only require a starting radius - generally to coincide with the wellbore radius r_w - and the angle to be subtended per grid block, $\Delta\theta$. From these two values, a grid structure can be devised to extend as far out radially as the simulation requires.

It is important to note that there will now be two overlaid grids which we will have to keep track of: A *node* grid and *face* grid. The node grid is a data set that stores

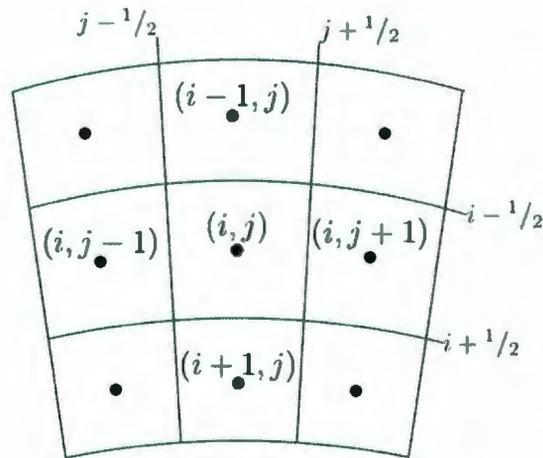


Figure 4.7: Areal view of a section of the cylindrical grid structure illustrating the numbering of grid blocks. Nodes are designated (i, j) where i is the radial location and j is the angular location. The interface locations are marked here with their respective $1/2$ increments.

the locations of all the nodes, or logarithmic centers, in r and θ of the grid blocks. As the face grid stores the radii and angle of the grid block boundaries. Each grid block now has an associated center node and boundaries, where the node represents the location of the block and its physical properties - e.g. pressure, permeability, etc - and the faces will be the interface where the fluid velocities are determined. The nodes, herein, are denoted in i and j and the faces being denoted by half-increments ahead or behind these, for example $i \pm 1/2$ or $j \pm 1/2$, as in Figure 4.7.

4.4 Cylindrical velocity discretization

In a cylindrical coordinate system, we must carefully organize the discretized flux equations for a cylindrical annular grid cell ensuring that each cell is referenced appropriately. Recalling from Section 3.1 the Darcy equations for the cylindrical coordinate

system are

$$\begin{bmatrix} u_r \\ u_\theta \\ u_z \end{bmatrix} = -\frac{1}{\mu} \begin{bmatrix} K_r & K_t & 0 \\ K_t & K_\theta & 0 \\ 0 & 0 & K_z \end{bmatrix} \begin{bmatrix} \partial_r P \\ 1/r \partial_\theta P \\ \partial_z P \end{bmatrix}, \quad (4.9)$$

where

$$\begin{aligned} K_r &= K_x \cos^2 \theta + K_y \sin^2 \theta \\ K_t &= (K_y - K_x) \sin \theta \cos \theta \\ K_\theta &= (K_x \sin^2 \theta + K_y \cos^2 \theta). \end{aligned}$$

These equations must be discretized in order to calculate the volumetric flux across each face of the grid blocks. The flux is determined based on the pressure and permeability values as calculated in the aforementioned TPFA.

4.4.1 Numerical gradients

Our pressure field from the TPFA calculation is arranged such that it is a matrix of values matching the dimensions of our node grid, where starting from a node $P_{i,j}$, a step to $P_{i+1,j}$ would indicate the step toward the wellbore in radius. Conversely, from $P_{i,j}$ to $P_{i-1,j}$ would step outward to a greater radius. Similarly, for the angular direction, from $P_{i,j}$ to $P_{i,j+1}$ will move clockwise in angle and to $P_{i,j-1}$, counter-clockwise, as illustrated in Figure 4.7.

To approximate the derivatives of pressure we will use a rudimentary approximation whereby

$$\frac{\partial P}{\partial r} \simeq \frac{P_{i+1,j} - P_{i,j}}{r_{i+1,j} - r_{i,j}} \quad \text{and} \quad \frac{\partial P}{\partial \theta} \simeq \frac{P_{i,j+1} - P_{i,j}}{\theta_{i,j+1} - \theta_{i,j}}.$$

This approximation of the partial derivatives will be adopted from here on. The

above examples are discretizations for the face at $r_{i+1/2,j}$ and $\theta_{i,j+1/2}$, as these faces lie between the radii and angles in the denominators of these expressions.

In the above expressions, we should note that we have off-diagonal entries in the permeability matrix denoted K_t in equation (4.9). These entries exist for anisotropic flow, but vanish to zero in isotropic flow as $K_y - K_x = 0$. However, we must include these portions into our discretization scheme. These off-diagonal terms now account for a portion of the angular flow in the u_r direction and a portion of radial flow in u_θ which is result of the anisotropy. For each boundary face of each block, we require a determination of the fluid velocity normal to that surface. Take for example, the face at a radius $r_{i+1/2}$. In order to approximate the velocity $u_{r+1/2}$, we require a determination of the angular term $-1/\mu r K_t \partial P / \partial \theta$. We note that this face extends along an arc from $\theta_{j-1/2}$ to $\theta_{j+1/2}$. To do so, a straight linear average of the approximated derivatives is taken. So, for the face at $r_{i+1/2}$, in block r_i , the off-diagonal gradient at face $\theta_{j-1/2}$ and $\theta_{j+1/2}$ becomes

$$\left(\frac{\partial P}{\partial \theta} \right)_t = \frac{1}{2} \left[\frac{P_{i,j+1} - P_{i,j}}{\theta_{i,j+1} - \theta_{i,j}} + \frac{P_{i,j} - P_{i,j-1}}{\theta_{i,j} - \theta_{i,j-1}} \right].$$

This represents the pressure gradient associated with the K_t term in the discretization of $u_{r+1/2}$.

4.4.2 Upscaling procedures

Now we must also consider the fluid mobilities λ between two adjacent grid-blocks. As we take two node pressures of adjacent blocks to determine the pressure gradient, we must also consider that each grid-block may have different fluid mobility values - recall, mobility refers to K/μ . When referring to an average of the fluid mobilities, the term *upscaling* is generally applied. In this process, the fluid mobilities are

approximated using various methods that are designed to reduce numerical dispersion statistically. Herein, we will adopt standard upscaling methods for a cylindrical geometry as outlined by [Aziz & Settari, 1979].

When two blocks are radially adjacent, where blocks r_{i+1} and r_i are considered, then the mobility associated to radius $r_{i+1/2}$ must be upscaled. For an interface that is vertically aligned as it is here, the logarithmic mean

$$\lambda_{i+1/2} = \frac{\ln(r_{i+1}/r_i)}{\frac{\ln(r_{i+1}/r_{i+1/2})}{\lambda_{i+1}} + \frac{\ln(r_{i+1/2}/r_i)}{\lambda_i}} \quad (4.10)$$

is employed.

In the angular direction, where the interface at $\theta_{j+1/2}$ is concerned, we instead use the common arithmetic mean,

$$\lambda_{j+1/2} = \frac{\lambda_{j+1} + \lambda_j}{2} . \quad (4.11)$$

Expressions (4.10) and (4.11) for the upscaling in r and θ will be used in the upcoming discretized equations quite extensively.

Similarly, following from the pressure gradient numerics, we need to account for the off-diagonal terms in the permeability matrix of equation (4.9) associated with the K_t/μ in u_r and u_θ . Again, we will employ a linear average value between the adjacent blocks where

$$\lambda_{t(i+1/2,j)} = \frac{1}{2} \left[\frac{\lambda_{j+1} + \lambda_j}{2} + \frac{\lambda_j + \lambda_{j-1}}{2} \right] = \frac{1}{2} \left[\frac{\lambda_{j+1} + 2\lambda_j + \lambda_{j-1}}{2} \right]$$

is the average for the tangential mobility term for a step in the r -direction, and

$$\lambda_{t(i,j+1/2)} = \frac{1}{2} \left[\frac{\ln(r_{i+1}/r_i)}{\frac{\ln(r_{i+1}/r_{i+1/2})}{\lambda_{i+1}} + \frac{\ln(r_{i+1/2}/r_i)}{\lambda_i}} + \frac{\ln(r_i/r_{i-1})}{\frac{\ln(r_i/r_{i-1/2})}{\lambda_i} + \frac{\ln(r_{i-1/2}/r_{i-1})}{\lambda_{i-1}}} \right]$$

is the average of the tangential mobility for a step in the θ -direction.

4.4.3 Discretized flow velocity expressions

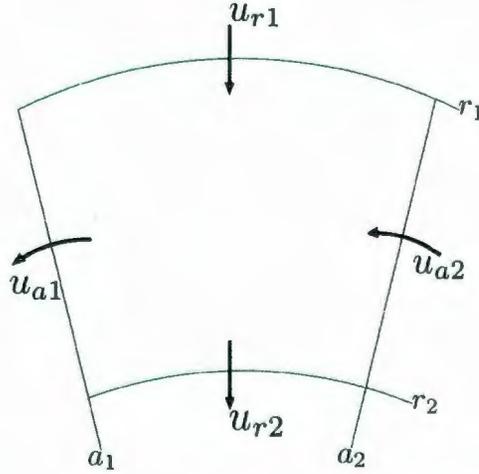


Figure 4.8: The identities of the sides of each grid block in the cylindrical block. The radius r_1 is the outermost boundary and r_2 the innermost. The angle $a_1 > a_2$ measured in the usual counter-clockwise direction. The velocities u are noted with the subscript identifying their respective face.

Employing, all of the approximations in Section 4.4 thus far, the following expressions are implemented to determine the velocity expressions needed. Taking into account the pressure and mobility discretizations as above, the full equations to be implemented become

$$u_{r1} = u_{i-1/2,j} = - \left(\frac{\ln(r_{i-1}/r_i)}{\frac{\ln(r_{i-1}/r_{i-1/2})}{\lambda_{i-1}} + \frac{\ln(r_{i-1/2}/r_i)}{\lambda_i}} \right) \left[\frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i} \right] \quad (4.12)$$

$$- \frac{1}{2} \left(\frac{\lambda_{j+1} + 2\lambda_j + \lambda_{j-1}}{2r_i} \right) \left[\frac{P_{i,j+1} - P_{i,j}}{\theta_{i,j+1} - \theta_{i,j}} + \frac{P_{i,j} - P_{i,j-1}}{\theta_{i,j} - \theta_{i,j-1}} \right]$$

and

$$u_{r2} = u_{i+1/2,j} = - \left(\frac{\ln(r_i/r_{i+1})}{\frac{\ln(r_i/r_{i+1/2})}{\lambda_i} + \frac{\ln(r_{i+1/2}/r_{i+1})}{\lambda_{i+1}}} \right) \left[\frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}} \right] \quad (4.13)$$

$$- \frac{1}{2} \left(\frac{\lambda_{j+1} + 2\lambda_j + \lambda_{j-1}}{2r_i} \right) \left[\frac{P_{i,j+1} - P_{i,j}}{\theta_{i,j+1} - \theta_{i,j}} + \frac{P_{i,j} - P_{i,j-1}}{\theta_{i,j} - \theta_{i,j-1}} \right]$$

for the two radial faces of the block illustrated in Figure 4.8, and

$$u_{a1} = u_{i,j-1/2} = - \left(\frac{\lambda_{j-1} + \lambda_j}{2r_i} \right) \left[\frac{P_{i,j-1} - P_{i,j}}{\theta_{j-1} - \theta_j} \right] \quad (4.14)$$

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

$$\left[\frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i} + \frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}} \right]$$

and finally

$$u_{a2} = u_{i,j+1/2} = - \left(\frac{\lambda_j + \lambda_{j+1}}{2r_i} \right) \left[\frac{P_{i,j} - P_{i,j+1}}{\theta_j - \theta_{j+1}} \right] \quad (4.15)$$

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

$$\left[\frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i} + \frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}} \right]$$

for the two angular faces of the block. These expressions now represent the numerical discretization for the Darcy velocity of an anisotropic medium, where the permeability field is reduced to its principal directions, K_x and K_y .

For an isotropic flow, the Darcy velocity equations are reduced to a more simple form, where the tangential permeability is $K_t = (K_y - K_x) \sin \theta \cos \theta$. We quickly see that in the case of $K_y = K_x$, $K_t = 0$. With that, the expressions for an isotropic permeability field reduce to

$$u_{r1} = u_{r(i-1/2,j)} = -\lambda_{r(i-1/2,j)} \frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i}$$

where $u_{r(i,j-1/2)}$ is the average velocity corresponding to the difference between center-node of the respective grid-block at a radius of $r_{i-1/2,j}$ and $r_{i,j}$. Similarly to the description for u_{r1} ,

$$u_{r2} = u_{r(i+1/2,j)} = -\lambda_{r(i+1/2,j)} \frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}}.$$

In the angular direction, designated by the subscript a , the flux equations for a grid cell are

$$\begin{aligned} u_{a1} &= u_{(i,j-1/2)} = -\lambda_{a(i,j-1/2)} \frac{P_{i,j} - P_{i,j-1}}{r_i \Delta\theta}, \\ u_{a2} &= u_{(i,j+1/2)} = -\lambda_{a(i,j+1/2)} \frac{P_{i,j+1} - P_{i,j}}{r_i \Delta\theta}. \end{aligned}$$

Although these expressions for the isotropic are much simplified, the actual coding uses only the discretized expressions given by 4.12-4.15. We should see that the isotropic expressions are in fact special cases of the anisotropic expressions, as the values of the off-diagonal terms vanish to zero.

4.5 Discretized TOF expressions

The time of flight (TOF) equations given in Sections 3.2.3 and 3.2.4 now must be discretized with the appropriate data. To begin, the coefficients of the velocity interpolations must be defined. These variables are then implemented in the TOF equations as previously given so that a time value can be determined.

4.5.1 Radial TOF

The coefficients of the velocity interpolation in the radial direction are defined as

$$A_r = \frac{u_{r2} - u_{r1}}{\frac{1}{r_2} - \frac{1}{r_1}} = \frac{r_1 r_2 (u_{r2} - u_{r1})}{r_1 - r_2}, \text{ and } B_r = u_{r1} + \frac{r_2 (u_{r2} - u_{r1})}{r_1 - r_2}.$$

The coefficients A_r and B_r are determined from the radial velocity interpolation expression, $u(r) = A_r/r(t) + B_r$, by simple algebra with the conditions that the velocities at radial interfaces are known and equal to $u_{r1} = A_r/r_1 + B_r$ and $u_{r2} = A_r/r_2 + B_r$.

The TOF for the travel-time to either radial interface (at r_1 or r_2) in the radial direction then becomes

$$t_r = \frac{1}{B_r} \left[(r_X - r_E) - \frac{A_r}{B_r} \ln \left[\frac{A_r + B_r r_X}{A_r + B_r r_E} \right] \right] \quad (4.16)$$

where t_r is the radial TOF, r_X the potential exit radius, and r_E the known entrance radius. With equation (4.16), for a known entrance radius r_E and face velocities u_{r1} at r_1 and u_{r2} at r_2 , the time it would take neutral particle to reach a potential exit radius at r_X can be calculated.

4.5.2 Angular TOF

Given the velocity interpolation as laid out by Pollock, we make a subtle alteration to the linear angular velocity interpolation, which is purely algebraic³, to become $u(\theta) = A_\theta \theta(t) + B_\theta$. From this interpolation, we can determine the constants

$$A_\theta = \frac{u_{a2} - u_{a1}}{\theta_2 - \theta_1}, \text{ and } B_\theta = u_{a1} - A_\theta \theta_1.$$

³As the Pollock interpolation was given as $u(\theta) = A_\theta(\theta(t) - \theta_1) + u_{\theta_1}$ previously, but on inspection we can see that this is identical to that laid out here; $u(\theta) = A_\theta \theta(t) + u_{\theta_1} - A_\theta \theta_1 = A_\theta \theta(t) + B_\theta$. It is altered in form here for a programming simplification.

Solving the TOF for the angular direction, yields

$$t_\theta = \frac{1}{A_\theta} \left(\ln \left[\frac{A_\theta \theta_X + B_\theta}{A_\theta \theta_E + B_\theta} \right] \right),$$

where, similarly, t_θ is the angular TOF, θ_X the potential exit angle, θ_E the known entrance angle. As before, given the entrance angle θ_E and the velocities u_{a1} at θ_1 and u_{a2} at θ_2 , the time to traverse the block angularly to θ_X can be determined.

4.6 Exit point determination

Once the TOF values are calculated, there are multiple values of travel-times through a grid block; time values are determined, based on the velocity profiles, from the entry face to the adjacent and opposite faces within the block. The times values can be negative, zero, or positive.

A negative TOF value indicates that the solution for a neutral particle to travel from point entrance to exit would have to be against the velocity direction, as only a solution backwards in time exists. In the formulation at hand, a zero is always returned if the entrance and the exit radius/angle coincide as both the expressions for the TOF will result in an expression with $\ln[1]$, which is identically zero. Also, for t_r , if the entrance and exit point are at equal radii, $r_X - r_E = 0$. Finally, the third option is a positive time value. This indicates that the tested exit face is a possibility as the travel-time is solution is forward in time for the velocity field. The least positive value indicates that, given the velocities for each face and the interpolation scheme outlined, a neutral particle would traverse to that side before hitting either of the others.

With the exit face determined from the least positive TOF, we can then go about determining the appropriate exit location within the cell. Given that we know the

velocities at each cell face, and that we have a velocity interpolation scheme whereby we have a piecewise continuous velocity in each direction, as well as the time taken to reach the exit face, we can therefore determine the distance the particle would in the perpendicular direction in that amount of time. From here, some simple laws of physics come into play. We will refer to the least positive TOF from the calculations as t_X in the following expressions.

4.6.1 Radial exit point

In the case where the least positive TOF, denoted t_X , indicates that the particle, entering at location (r_E, θ_E) , would exit at either angular face of θ_1 or θ_2 , then we must calculate the displacement in the radial direction to determine the exit radius r_X . Knowing the time that a particle would spend traversing the grid-block, a displacement is calculated.

In view of expression (4.16), the TOF t_r , we see that there is no algebraic solution for r_X using standard mathematical functions⁴ and consequently, an approximation must be made in order to keep this code numerically efficient. As previously derived in Section 3.2.3, we employ kinematics to approximate the radial displacement for the determined time step. From the initial derivation for the kinematics expression for the exit radius r_X , we would have

$$r_X = r_E + u_E t_X - \frac{1}{2} \left(\frac{A_r}{r^2} u(r) \right) t_X^2 ,$$

where u_E is the initial velocity at the entrance face. We note, that the acceleration

⁴There exists an analytical solution to the radial time of flight equation (4.16) for the radial exit point r_X by using the Lambert W function, sometimes referred to as the Omega function or the product logarithm. It is an inverse function of $f(w) = we^w$ where e is the exponential function and w is a complex number. It however, cannot be expressed using elementary functions and hence, is of little use in a discretization scheme.

term

$$a(t) = \frac{A_r}{r^2} u(r)$$

must be approximated as both r and $u(r)$ are changing through the block. We will assume that the acceleration through the block is roughly constant and set the acceleration to be the that of the node radius, and hence the radial exit displacement becomes

$$r_X = r_E + u_E t_X - \frac{1}{2} \left(\frac{A_r}{r_N^2} u_N \right) t_X^2, \quad (4.17)$$

where r_N is the radius of the node for the block in which we are simulating and $u_N = A_r/r_N + B_r$.

4.6.2 Angular exit point

If instead we find that the least positive TOF t_X is at one of the radial faces r_1 or r_2 , then we must calculate the displacement in the angular direction a_X .

From Section 3.2.4, we found that the angular expression can be solved algebraically and given as

$$\theta_X = \frac{u_E}{A_\theta} [\exp(A_\theta t_X) - 1] + \theta_E. \quad (4.18)$$

Here, u_E is the velocity at entrance face θ_E .

4.7 General code structure

The process of determining streamline's through a grid block and, in turn, through the medium, requires a logical loop structure to navigate and assemble the discretized equations properly. To see the actual code used herein, the reader is directed to Appendix C. The actual logic is displayed in a flow chart illustrated in Figure 4.9.

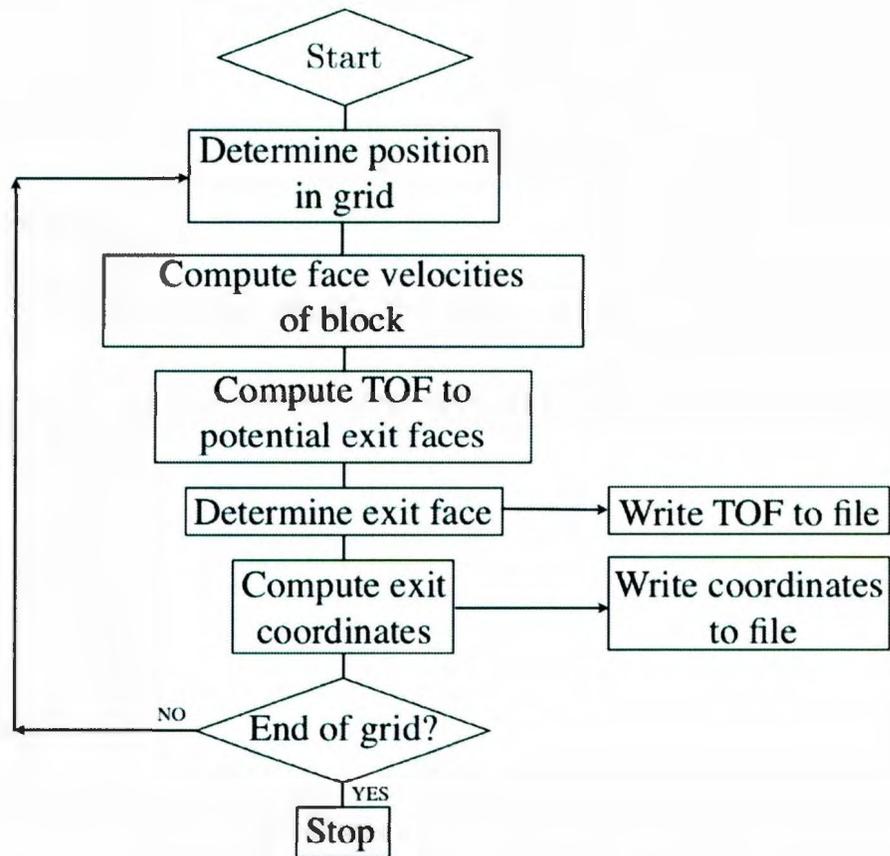


Figure 4.9: Flow chart showing the logic structure for streamline generation.

Chapter 5

Results and Conclusions

Implementation of the code was chosen to be written for a *quarter-five spot* pattern. This well structure is commonly chosen for testing theory in reservoir simulation. As a “normal” well distribution with a permeability field reduced to its principal components, the pressure distribution will be symmetric in the four areal quadrants. Therefore, this effectively reduces the computational requirements to a quarter of what is otherwise necessary.

Here, it was chosen to have an injection site (source) of arbitrary volumetric flow in the top right corner, and a production site (sink) at the bottom left corner. The sink/source terms, for each grid block, are otherwise set to zero for no-flow boundaries. Hence, all flow entering via the source must exit the sink. This produces clearly defined pressure contours that are easily understood, and make it easy to predict the direction of flow, which in turn, predict the likely path of each streamline. In general, the streamlines should flow from high pressure to low pressure and remain, given normal permeability parameters, roughly perpendicular to the pressure contours.

As the discretization scheme in the previous chapter is designed, a pressure value is required directly perpendicular to either side of a grid face; a face sitting at point

n	$r_n[m]$	i	$r_i[m]$
1	0.1000	1	0.10849
2	0.1175	2	0.12743
3	0.1380	3	0.14967
4	0.1620	4	0.17579
5	0.1903	5	0.20647
6	0.2235	6	0.24251
7	0.2625	7	0.28483
8	0.3084	8	0.33455
9	0.3622	9	0.39294
10	0.4254	10	0.46152
11	0.4996	11	0.54207
12	0.5868	12	0.63668
13	0.6893	13	0.7478
14	0.8096	14	0.87831
15	0.9509	15	1.0316
16	1.1168	16	1.2117
17	1.3117	17	1.4231
18	1.5407	18	1.6715
19	1.8096	19	1.9632
20	2.1254	20	2.3059
21	2.4963	21	2.7083
22	2.9320	22	3.181
23	3.4438	23	3.7362
24	4.0448	24	4.3883
25	4.7508	25	5.1543
26	5.5799	26	6.0538
27	6.5538	27	7.1104
28	7.6977	28	8.3514
29	9.0412	29	9.809
30	10.6192		

Table 5.1: Table of radial grid boundaries r_n on the left and node locations r_i on the right as calculated from starting radius of $r = 0.1m$ and determined using equations (4.7) and (4.8).

$(i - 1/2, j)$, requires a pressure value to exist at both points $(i - 1, j)$ and (i, j) for a proper velocity determination. A consequence of having this discretization scheme, is that we must have at least one pressure value on all sides of the grid-block we wish to simulate. Hence, a point *within* the radial grid must be chosen as the originating position of the streamline.

For the following simulations, an initial wellbore radius of $r = 0.1m$ was chosen as the starting point. Based on this and in view of equation (4.7), we are able to calculate the first 30 radial steps as given in Table (5.1).

With the radii of the radial grid boundaries, we are also free to calculate the node location radii which are given by the logarithmic mean of successive grid boundaries given in equation (4.8). The calculated means between boundaries for the nodes are explicitly written in Table (5.1) and we note that the num-

ber of i -values is therefore one less than the number of n -values. In summary, 29 radial grid blocks consist of 30 boundary radii to create the closed grid system.

For the angular steps, it was chosen to span ten degrees ($10^\circ \approx 0.1745$ radians) for each block. This makes each grid block's angular boundary easily determined as there will be 9 grid blocks, and 10 grid boundaries, in the angular direction and since the velocity interpolation scheme is a standard average, the geometric middle angle of each block is necessary. These are given in Table (5.2) .

m	$\theta_m[rad]$	j	$\theta_j[rad]$
1	0	1	0.0873
2	0.1745	2	0.2618
3	0.3491	3	0.4363
4	0.5236	4	0.6109
5	0.6981	5	0.7854
6	0.8727	6	0.9599
7	1.0472	7	1.1345
8	1.2217	8	1.3090
9	1.3963	9	1.4835
10	1.5708		

Table 5.2:

5.1 Pressure calculation for quarter five-spot configuration

By implementing the discretization as laid out in Section 4.1 we use a standard *two-point flux approximation* (TPFA) to solve for the pressure distribution on a standard Cartesian grid. With the Cartesian grid set at 501 by 501, we set the flux discharge on each grid cell as being zero everywhere except at the top right and bottom left of the grid with a value of $q_{in} = 1 \times 10^{-3} m^3/s$ and $q_{out} = -1 \times 10^{-3} m^3/s$, respectively. These flux values now represent an injector-producer pair of wells within the medium. In the one case, the injector, introduces an influx (source) of fluid and in the other, the producer, provides an outlet (sink). With equivalent magnitudes of flux, and no flow at any outermost boundaries, the resulting flow will be mass-conserving.

The pressure data for streamline simulation is extracted from the Cartesian pressure distribution by locating the pressure value nearest the node position in the overlaid radial grid. In doing so, the actual pressure data used to calculate the streamlines is reduced from a 501 by 501 Cartesian grid, to a matrix of 29 by 9. Pressure values that result are on the order of $10 MPa = 10^7 Pa$ and are oriented such that pressure value located at (1, 1) corresponds to the top, left-hand node located at $r = 9.8090m$ and $\theta = 1.4835$ radians and pressure (29, 9) is nearest the producing well very near the horizontal axis in the lower left at $r = 0.10848m$ and $\theta = 0.0873$ radians.

In short, the structure of the pressure data is such that the radius is constant along rows of the matrix, and the angle is constant along the columns. This organization makes the streamline simulation calculations highly intuitive.

For clarity, in the forthcoming sections, the radial grid employed for the streamline simulation will be overlaid onto a pressure contour plot to make the geometry more digestible to the reader. The pressure node locations are illustrated by the white

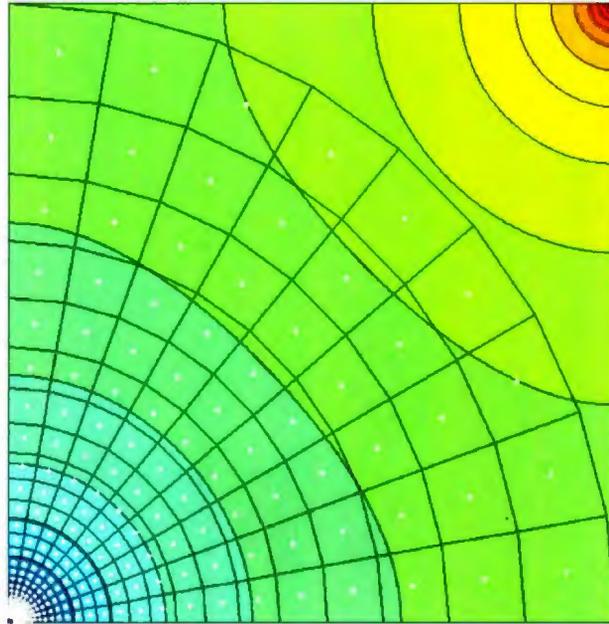


Figure 5.1: Isotropic pressure contours with radial grid overlay. Note the injection site at the top right, noted by a high pressure in red, and the production site in the bottom left corner where the colour turns blue denoting a low pressure. The radial grid system is overlaid illustrating the block dimensions.

points roughly in the center of each radial grid block.

5.1.1 Isotropic pressure calculation

The TPFPA calculation is used to calculate the pressure values on a 501 by 501 Cartesian grid with an arbitrary fluid mobility value of $\lambda = \mathbf{K}[m^2]/\mu[Pa \cdot s] = 0.5 \times 10^9 m^3/kg \cdot s$ which is roughly equivalent to the mobility of water in an aquifer. We note here that this is an isotropic system, so there is no variability introduced for the fluid mobility in terms of direction; the mobility in the x and y directions are equivalent.

The shape of the pressure contours in Figure 5.1 are, as to be expected, circular near both wells. This is best observed near the producing well in the lower left where the pressure contours are more easily compared with the overlaid radial grid. We see that the pressure at equal radii are roughly equivalent near each well, however, we

P (MPa)	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$
$i=1$	7.9925	8.093	8.2643	8.4465	8.5349	8.4465	8.2643	8.093	7.9925
$i=2$	7.9003	7.9579	8.0497	8.1434	8.1805	8.1434	8.0497	7.9579	7.9003
$i=3$	7.7578	7.7876	7.836	7.8841	7.9058	7.8841	7.836	7.7876	7.7578
$i=4$	7.585	7.6012	7.6265	7.6496	7.6658	7.6496	7.6265	7.6012	7.585
$i=5$	7.3989	7.4056	7.4229	7.4344	7.4424	7.4344	7.4229	7.4056	7.3989
$i=6$	7.2037	7.2095	7.2133	7.2229	7.2241	7.2229	7.2133	7.2095	7.2037
$i=7$	7.0013	7.0078	7.0067	7.0141	7.0116	7.0141	7.0067	7.0078	7.0013
$i=8$	6.7999	6.8068	6.807	6.8108	6.8101	6.8108	6.807	6.8068	6.7999
$i=9$	6.5986	6.5988	6.6063	6.5988	6.6014	6.5988	6.6063	6.5988	6.5986
$i=10$	6.3935	6.3982	6.3946	6.3979	6.4032	6.3979	6.3946	6.3982	6.3935
$i=11$	6.1913	6.1902	6.1971	6.1984	6.1886	6.1984	6.1971	6.1902	6.1913
$i=12$	5.983	5.9886	5.9849	5.9846	6.0002	5.9846	5.9849	5.9886	5.983
$i=13$	5.7909	5.7907	5.7905	5.7862	5.7792	5.7862	5.7905	5.7907	5.7909
$i=14$	5.5867	5.5837	5.5906	5.5949	5.5761	5.5949	5.5906	5.5837	5.5867
$i=15$	5.3697	5.3872	5.3883	5.3909	5.372	5.3909	5.3883	5.3872	5.3697
$i=16$	5.1732	5.1841	5.1622	5.1906	5.1726	5.1906	5.1622	5.1841	5.1732
$i=17$	4.9731	4.9771	4.9851	4.9822	4.9871	4.9822	4.9851	4.9771	4.9731
$i=18$	4.7819	4.782	4.7794	4.7915	4.7699	4.7915	4.7794	4.782	4.7819
$i=19$	4.5517	4.5985	4.5785	4.5951	4.5786	4.5951	4.5785	4.5985	4.5517
$i=20$	4.394	4.3841	4.3917	4.3767	4.3534	4.3767	4.3917	4.3841	4.394
$i=21$	4.1484	4.191	4.2013	4.1913	4.1775	4.1913	4.2013	4.191	4.1484
$i=22$	3.9964	3.9632	3.978	3.9924	3.9735	3.9924	3.978	3.9632	3.9964
$i=23$	3.7433	3.7771	3.7905	3.8097	3.7304	3.8097	3.7905	3.7771	3.7433
$i=24$	3.5419	3.5879	3.5706	3.5976	3.5889	3.5976	3.5706	3.5879	3.5419
$i=25$	3.4281	3.3694	3.4176	3.4409	3.4298	3.4409	3.4176	3.3694	3.4281
$i=26$	3.1654	3.1992	3.2467	3.2625	3.2479	3.2625	3.2467	3.1992	3.1654
$i=27$	3.0113	3.0542	2.9682	3.0559	3.0356	3.0559	2.9682	3.0542	3.0113
$i=28$	2.837	2.893	2.8107	2.8107	2.7809	2.8107	2.8107	2.893	2.837
$i=29$	2.5947	2.6369	2.7127	2.641	2.7809	2.641	2.7127	2.6369	2.5947

Table 5.3: Pressure data for *isotropic* radial grid. (i, j) indicates $(P_{radial}, P_{angular})$ location in grid according to the radial and angular node locations.

note that the pressure contours near the center of the plot begin to “flatten” as the curves of the contours alter their curvature from one well to the other. Again this is mostly easily observed in comparison to the overlaid radial grid. It is evident from this plot that the presence of each well affects the pressure distribution throughout the medium, which may be obvious, but is an important result. Given that the pressure distribution is altered by the presence of more wells, this then indicates that the direction of fluid motion is altered as we know that the direction of flow which is tangent to the velocity vector for an isotropic medium is dictated by the direction of the negative gradient of pressure, e.g. Darcy’s Law.

We should expect, based on the pressure contours, that the pressure in the isotropic medium to be symmetric about a diagonal connecting the two wells. If we look carefully at Table 5.3 in relation to Figure 5.1, we should see that the pressure at the nodes are symmetric about column $j = 5$; $P_{j=4} = P_{j=6}$; $P_{j=3} = P_{j=7}$ and so on. We also see that as we approach the producing well in the lower left corner, at for example row $i = 7$, there is very little variance in the pressure along this radius; at maximum the pressure only varies by $\pm 0.15\%$ along radius $i = 7$. We would also note that at $i = 1$, the largest radius from the producing well, the pressure values are the highest, and at $i = 29$, closest to the producing wellbore, the pressures are the lowest. Meaning, the flow of fluid, being from high pressure to low, is in the direction of the producing well.

These results are in line with what we would expect from our pressure calculation and this data will be used as the basis of our streamline simulation for the isotropic case.

P (MPa)	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$
$i=1$	10.684	10.802	10.997	11.185	11.224	11.031	10.739	10.481	10.331
$i=2$	10.551	10.611	10.699	10.771	10.756	10.643	10.468	10.313	10.219
$i=3$	10.347	10.37	10.402	10.419	10.393	10.309	10.199	10.102	10.045
$i=4$	10.103	10.109	10.113	10.103	10.076	10.006	9.9332	9.8689	9.8309
$i=5$	9.8443	9.8388	9.8351	9.8141	9.7816	9.7273	9.674	9.6223	9.5976
$i=6$	9.5762	9.5714	9.5518	9.5305	9.4938	9.4532	9.4044	9.3724	9.3505
$i=7$	9.3013	9.2977	9.274	9.2528	9.214	9.1807	9.1377	9.1144	9.092
$i=8$	9.0298	9.0278	9.0074	8.9833	8.9487	8.915	8.8786	8.8551	8.8331
$i=9$	8.7603	8.7498	8.74	8.7023	8.674	8.638	8.6179	8.5854	8.5727
$i=10$	8.4872	8.4831	8.4587	8.4375	8.4134	8.3744	8.3422	8.3245	8.3064
$i=11$	8.2186	8.207	8.1979	8.1742	8.1312	8.1131	8.0836	8.0536	8.0431
$i=12$	7.9428	7.9409	7.9176	7.8919	7.8836	7.8334	7.806	7.7897	7.7713
$i=13$	7.6891	7.6797	7.6619	7.6318	7.5931	7.5718	7.5509	7.5308	7.5203
$i=14$	7.4198	7.406	7.3982	7.3789	7.3263	7.322	7.2893	7.2608	7.2529
$i=15$	7.1339	7.1467	7.1302	7.1109	7.0581	7.0542	7.0257	7.0036	6.9684
$i=16$	6.8748	6.8795	6.8345	6.8469	6.7962	6.7919	6.7273	6.7372	6.7114
$i=17$	6.6118	6.6079	6.6005	6.5745	6.5525	6.5166	6.4959	6.4649	6.4485
$i=18$	6.36	6.3504	6.3287	6.3248	6.2671	6.2653	6.2274	6.2099	6.1983
$i=19$	6.0578	6.1087	6.0648	6.0631	6.0158	6.011	5.9635	5.9696	5.8954
$i=20$	5.8502	5.8263	5.822	5.7808	5.72	5.7194	5.7154	5.6889	5.6889
$i=21$	5.5266	5.5729	5.5696	5.5306	5.489	5.483	5.4679	5.4348	5.3675
$i=22$	5.3282	5.2741	5.273	5.273	5.221	5.2176	5.1782	5.1351	5.166
$i=23$	4.9951	5.0318	5.0264	5.0295	4.9017	4.9815	4.9323	4.8881	4.8344
$i=24$	4.7298	4.78	4.7371	4.746	4.7158	4.7081	4.6439	4.6436	4.5708
$i=25$	4.5798	4.4876	4.541	4.5424	4.5068	4.4997	4.4375	4.3629	4.4219
$i=26$	4.2332	4.27	4.3223	4.3112	4.2678	4.2623	4.2063	4.1323	4.0788
$i=27$	4.0294	4.0763	3.944	4.0437	3.989	3.9867	3.8547	3.9456	3.878
$i=28$	3.7985	3.86	3.727	3.727	3.6544	3.6587	3.6587	3.7392	3.6514
$i=29$	3.4869	3.5326	3.6164	3.4901	3.6544	3.4508	3.51	3.3922	3.3252

Table 5.4: Pressure data for anisotropic radial grid. (i, j) indicates $(P_{radial}, P_{angular})$ location in grid according to the radial and angular node locations.

5.1.2 Anisotropic pressure calculation

For the anisotropic pressure calculation, the setup is similar to that of the isotropic case in the previous section, with only a few important distinctions. As previous, a 501 by 501 Cartesian grid was used as the basis of the pressure calculation. However, the mobility value is altered to be different for the x and y directions. Here, we've chosen the $\lambda_x = 0.69\lambda = 0.345 \times 10^{-9} \text{m}^3/\text{kg}\cdot\text{s}$ and $\lambda_y = 0.84\lambda = 0.42 \times 10^{-9} \text{m}^3/\text{kg}\cdot\text{s}$. This, in turn, means that fluid can more easily flow in the vertical y -direction than in the horizontal x -direction.

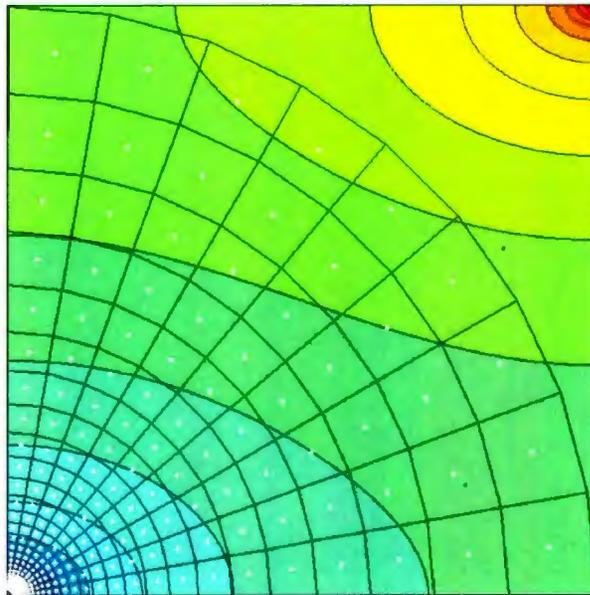


Figure 5.2: Anisotropic pressure contours with radial grid overlay.

As a result, we see in Figure 5.2 that the pressure contours are closer together in the vertical axes than the horizontal, which indicates that the gradient is of a greater magnitude in the vertical direction and this will therefore be the preferred direction of flow.

As before we note that the center of curvature of the pressure contours encircle each well, however, we note that the shape is now, quite plainly, elliptical as opposed

to circular. This is congruent with the anisotropic contours as discussed earlier. We see from Figure 5.2 that the pressure at equal radii are now no longer symmetric about the diagonal connecting the wells, as in the isotropic case. We now see that the pressure values near $j = 1$ remain higher than those at $j = 9$ for a given radius, i . In view of Table 5.4 we see that this is indeed the case. Take, for example, the pressure at node $(3, 1) = 10.347MPa$ is roughly equivalent to that at $(1, 9) = 10.331MPa$.

Given the pressure distribution, as calculated in Table 5.4, we would expect that the vertical component of the velocity to be of higher magnitude than that of the horizontal, as again, we must recall that the average velocity vector is proportional to the negative gradient of pressure. Although, in contrast to the isotropic case, the velocity vector in the anisotropic medium is generally not orthogonal to the pressure gradient as the coefficients of the fluid mobility, namely, $\lambda = \mathbf{K}/\mu$, are no longer equal for each term of the gradient.

5.2 Streamline results

For the streamline calculation, velocity values must be calculated for each grid boundary - referred to as a grid face - so that a velocity profile can be introduced and a time-of-flight (TOF) determined. Recall from Section 4.4.3, velocity values are determined for each face of the grid based on the pressure node values immediately opposed on either side of the face. Each boundary is shared by exactly two grid blocks, and each grid block has an associated pressure value for the block - where the pressure at the node is assumed to be homogeneous throughout each block - and so, the pressure gradients are determined at each face based on the pressures of the adjacent grid blocks. We should also recall that the permeability values for the adjacent blocks are upscaled in the standard method for this geometry as laid out by

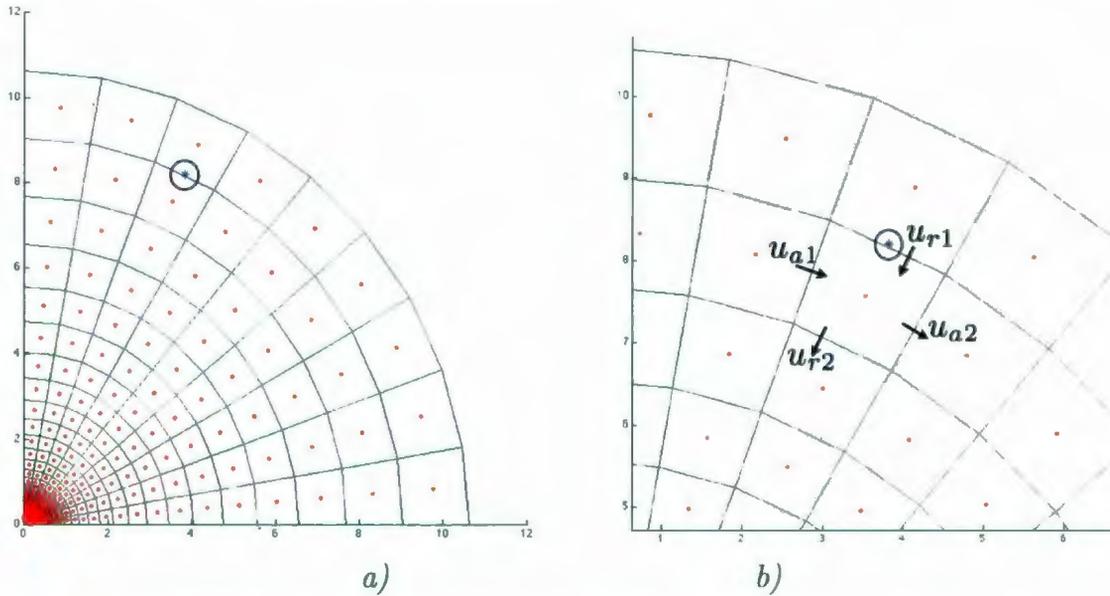


Figure 5.3: Plot of radial grid with start location and velocity vectors. Plot *a*) is of the entire grid with start location circled. This point is used for the beginning location of the test of the discretized code in the isotropic and anisotropic case. Plot *b*) is a close-up of the starting location and the first grid block. The velocities u_{r1} , u_{r2} , u_{a1} , and u_{a2} are illustrated.

[Aziz & Settari, 1979] whereby the radial gradients have logarithmic means employed for the permeabilities and the angular gradients have arithmetic means.

For the streamline calculation itself, a specific routine is followed. The streamline is originated within the radial grid at some point (i, j) so as to ensure that there are always pressure node values at $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$, and $(i, j + 1)$. With adjacent pressure values for the grid block where the process is started, face velocities are calculated. These face velocities are used to calculate the TOF - as derived from the velocity interpolation - for a neutral particle to travel from the entrance point to each possible exit face. With the calculated times, the least, positive time value indicates the face that the particle will exit on. Using this time value, the displacement in r and θ can then be determined and this is our block exit point. This exit is now carried forward to be the entrance location for the adjacent block and the routine begins again with determination of the face velocities from the surrounding pressure

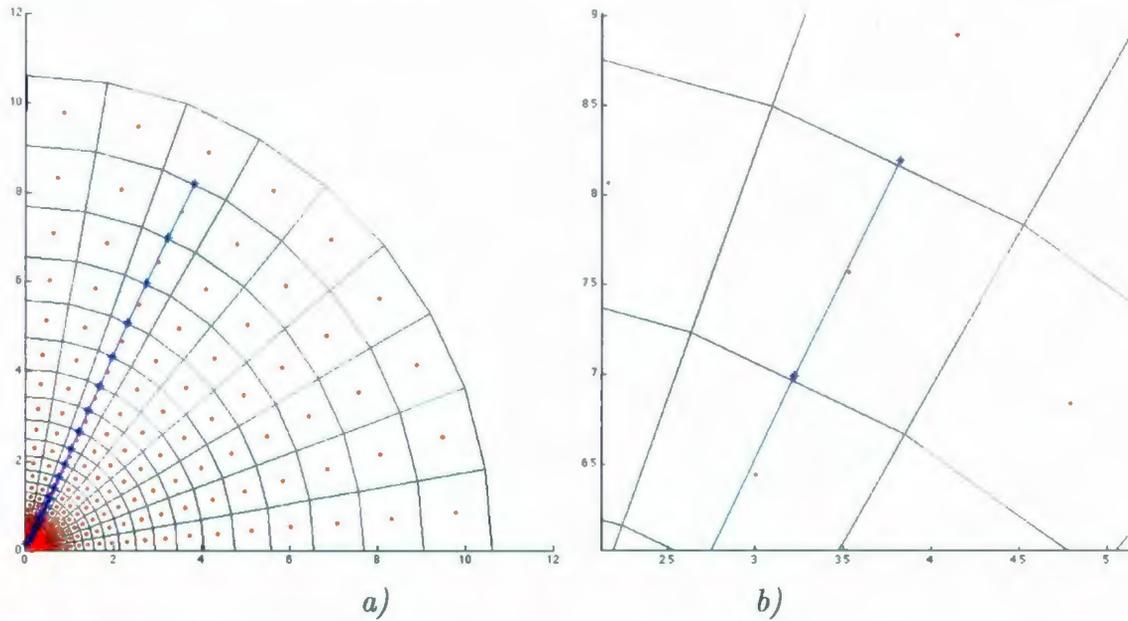


Figure 5.4: For an isotropic system. Plot *a)* shows the streamline originating from outer radius and traveling towards the wellbore. Plot *b)* is a close-up of the streamline's trajectory moving roughly orthogonal to the radius contours.

values. This process is repeated until the routine encounters an edge of the pressure data. Given the flow regime as it is, it should be near the smallest radial value of $r = 0.1175m$ close to the producing well radius.

5.2.1 Isotropic Streamline

To begin, a start location of $(r_0, \theta_0) = (9.0412, 1.1345)$ was chosen. This point is on the outermost face of the second radial ring of grid blocks. This starting point therefore satisfies having a pressure value available on all sides so that the velocity profile can be established. The velocities for all faces may now be calculated for this block.

The face velocities for the first block in the isotropic case are

$$u_{r1} = u_{r(i-1/2,j)} = -\lambda_{r(i-1/2,j)} \frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i} = -4.633 \times 10^{-5} m/s ,$$

$$\begin{aligned}
u_{r2} = u_{r(i+1/2,j)} &= -\lambda_{r(i+1/2,j)} \frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}} = -6.8619 \times 10^{-5} m/s , \\
u_{a1} = u_{(i,j-1/2)} &= -\lambda_{a(i,j-1/2)} \frac{P_{i,j} - P_{i,j-1}}{r_i \Delta\theta} = 1.6320 \times 10^{-6} m/s , \\
u_{a2} = u_{(i,j+1/2)} &= -\lambda_{a(i,j+1/2)} \frac{P_{i,j+1} - P_{i,j}}{r_i \Delta\theta} = 2.3682 \times 10^{-6} m/s ,
\end{aligned}$$

where $\lambda = \mathbf{K}/\mu$ and the subscript r indicates a radial logarithmic mean and the a indicates and angular arithmetic mean between mobilities. We note that the velocities for u_{a1} and u_{a2} in the angular direction are an order of magnitude smaller than those in the radial direction, namely, u_{r1} and u_{r2} . We should therefore expect the streamline to travel through this block and exit side r_2 towards the production well. We would also expect that the ratio of radial to angular displacement to be in the vicinity of 25 : 1 based on these velocities. A negative velocity value for u_{r1} and u_{r2} indicate that their directions are *towards* the producing well and the positive velocity values for u_{a1} and u_{a2} indicate that they are counter-clockwise with increasing angle.

As we see in Table 5.5, the first row of data indicates the starting position within the grid. Reflective of this, the time of flight (TOF) and the cumulative TOF begin at zero. In the second row, we see the very first exit location determined. We see that the time to traverse from a radius of 9.0412m to the exit radius at 7.6977m takes a total of 16930s \approx 4.7hrs. Given that the velocities are on the order of $10^{-5}m/s$, this value seems appropriate. We also note that the radial distance covered is \approx 1.34m and in that amount of radial distance an angle of $4.4 \times 10^{-3}rad$ is traversed. At an approximate radius of 8.3m, this gives a physical arc of 0.037m which would be in the correct range recognizing that the arc is different for each radius. We can plainly see that the flow is dominated by radial flow, which was to be expected.

This process is now carried forward, the new (r, θ) value becomes the entrance point within the adjacent block and new velocities are calculated, the TOF's deter-

Radius (m)	θ (rad)	TOF (sec)	CTOF (sec)
9.0412	1.1345	0	0
7.6977	1.1389	16930	16930
6.5538	1.1432	14761	31691
5.5799	1.1448	9220.7	40911
4.7508	1.1471	7052.4	47964
4.0448	1.1479	4809.4	52773
3.4438	1.1469	3603.9	56377
2.932	1.1474	2628	59005
2.4963	1.1496	1865.4	60870
2.1254	1.1501	1350	62220
1.8096	1.1532	977.88	63198
1.5407	1.153	714.09	63912
1.3117	1.153	520.49	64433
1.1168	1.1546	379.51	64812
0.95085	1.1483	277.01	65089
0.80956	1.1598	199.24	65289
0.68926	1.1661	144.51	65433
0.58684	1.1722	103.62	65537
0.49963	1.111	77.09	65614
0.42539	1.0934	58.732	65673
0.36218	1.0571	38.31	65711
0.30836	1.073	28.332	65739
0.26254	1.0429	22.481	65762
0.22352	1.0633	16.634	65778
0.19031	1.0874	13.716	65792
0.16203	1.0761	10.192	65802

Table 5.5: Isotropic streamline data. *Radius* and θ column indicate the locations a neutral particle would encounter at the grid boundaries as it passes through the medium in meters and radians, respectively. *TOF* is the time of flight of each step and the *CTOF* is the cumulative time of flight for the entire streamline measured in *seconds*.

mined, and these values used to determine the exit point. In view of the full streamline for the isotropic case, Figure 5.4, we see the full trajectory of the streamline simulation through the medium. We see that the streamline generally flows directly into the producing well with very little deviation in the angular direction. From Table 5.5, we see that the overall variation in the θ column is very slight, by comparison to the *Radial* column, with only a total variance of $0.022rad$ over the entire traverse from $r_0 = 9.0412m$ to $r = 0.16203m$.

As the equations for the isotropic case in two-dimensions would indicate, namely

$$\begin{bmatrix} u_r \\ u_\theta \end{bmatrix} = -\frac{1}{\mu} \begin{bmatrix} K_x \cos^2 \theta + K_y \sin^2 \theta & (K_y - K_x) \sin \theta \cos \theta \\ (K_y - K_x) \sin \theta \cos \theta & (K_x \sin^2 \theta + K_y \cos^2 \theta) \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial r} \\ 1/r \frac{\partial P}{\partial \theta} \end{bmatrix}, \quad (5.1)$$

the off-diagonal values for the permeability vanish to zero when $K_x = K_y$, as the off-diagonal terms are proportional to $(K_y - K_x)$. So, in the isotropic case, when $\mathbf{K} = K = \text{constant}$, the Darcy equation reduces to

$$\begin{bmatrix} u_r \\ u_\theta \end{bmatrix} = -\frac{1}{\mu} \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial r} \\ 1/r \frac{\partial P}{\partial \theta} \end{bmatrix}$$

and we see that the velocity components are dependent only on the respective gradient, e.g. $u_r \propto \partial_r P$.

5.2.2 Anisotropic streamline

As before, the start location for the anisotropic streamline simulation is chosen as $(r_0, \theta_0) = (9.0412, 1.1345)$. By choosing this point we can detail the differences between the two simulations and compare. The only distinction then between these two simulations will be the introduction of variability in the mobility term, which in turn,

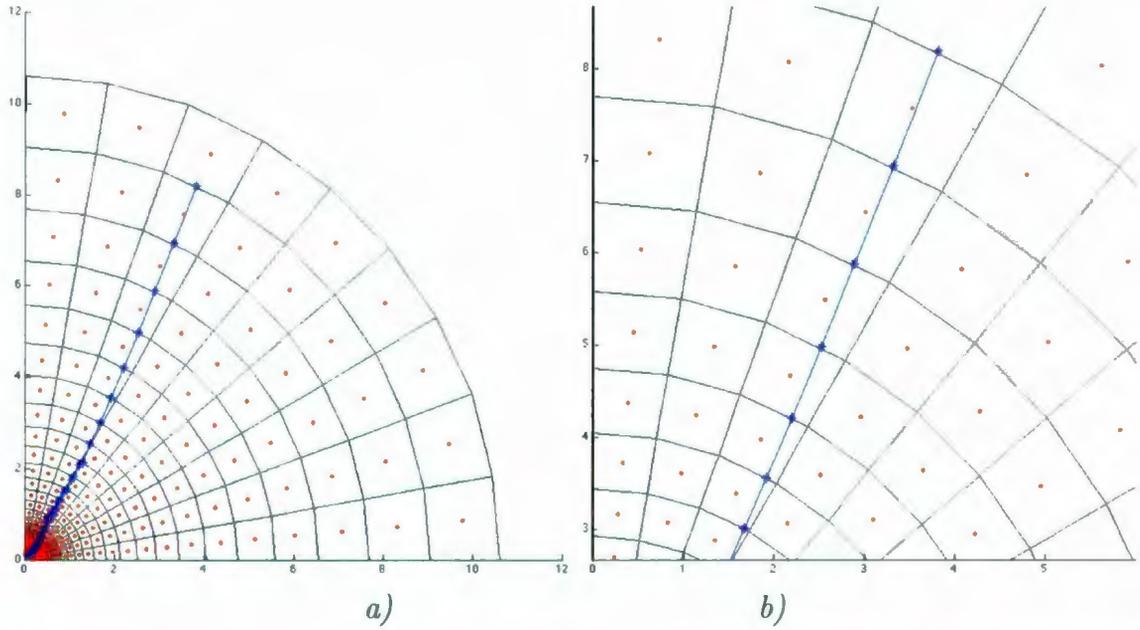


Figure 5.5: For an anisotropic system. Plot *a*) shows the streamline originating from an outer radius and travels toward the wellbore, but with a distinct curvature. Plot *b*) is a close-up of the streamline's initial trajectory illustrating an influence due to a more prominent angular flow.

affects the velocity equations quite drastically, which we will now discuss.

As before, the face velocities for each grid block are

$$\begin{aligned}
 u_{r1} = u_{i-1/2,j} &= - \left(\frac{\ln(r_{i-1}/r_i)}{\frac{\ln(r_{i-1}/r_{i-1/2})}{\lambda_{i-1}} + \frac{\ln(r_{i-1/2}/r_i)}{\lambda_i}} \right) \left[\frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i} \right] \\
 &\quad - \frac{1}{2} \left(\frac{\lambda_{j+1} + 2\lambda_j + \lambda_{j-1}}{2r_i} \right) \left[\frac{P_{i,j+1} - P_{i,j}}{\theta_{i,j+1} - \theta_{i,j}} + \frac{P_{i,j} - P_{i,j-1}}{\theta_{i,j} - \theta_{i,j-1}} \right] \\
 &= -8.2113 \times 10^{-5} \text{ m/s} ,
 \end{aligned}$$

and

$$\begin{aligned}
 u_{r2} = u_{i+1/2,j} &= - \left(\frac{\ln(r_i/r_{i+1})}{\frac{\ln(r_i/r_{i+1/2})}{\lambda_i} + \frac{\ln(r_{i+1/2}/r_{i+1})}{\lambda_{i+1}}} \right) \left[\frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}} \right] \\
 &\quad - \frac{1}{2} \left(\frac{\lambda_{j+1} + 2\lambda_j + \lambda_{j-1}}{2r_i} \right) \left[\frac{P_{i,j+1} - P_{i,j}}{\theta_{i,j+1} - \theta_{i,j}} + \frac{P_{i,j} - P_{i,j-1}}{\theta_{i,j} - \theta_{i,j-1}} \right]
 \end{aligned}$$

$$= -9.5759 \times 10^{-5} m/s ,$$

for the two radial faces of the block, and for the angular faces

$$\begin{aligned} u_{a1} = u_{i,j-1/2} &= - \left(\frac{\lambda_{j-1} + \lambda_j}{2r_i} \right) \left[\frac{P_{i,j-1} - P_{i,j}}{\theta_{j-1} - \theta_j} \right] \\ &\quad - \frac{1}{2} \left(\frac{\ln(r_{i+1}/r_i)}{\frac{\ln(r_{i+1}/r_{i+1/2})}{\lambda_{i+1}} + \frac{\ln(r_{i+1/2}/r_i)}{\lambda_i}} + \frac{\ln(r_i/r_{i-1})}{\frac{\ln(r_i/r_{i-1/2})}{\lambda_i} + \frac{\ln(r_{i-1/2}/r_{i-1})}{\lambda_{i-1}}} \right) \\ &\quad \left[\frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i} + \frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}} \right] \\ &= -4.6560 \times 10^{-6} m/s , \end{aligned}$$

and finally

$$\begin{aligned} u_{a2} = u_{i,j+1/2} &= - \left(\frac{\lambda_j + \lambda_{j+1}}{2r_i} \right) \left[\frac{P_{i,j} - P_{i,j+1}}{\theta_j - \theta_{j+1}} \right] \\ &\quad - \frac{1}{2} \left(\frac{\ln(r_{i+1}/r_i)}{\frac{\ln(r_{i+1}/r_{i+1/2})}{\lambda_{i+1}} + \frac{\ln(r_{i+1/2}/r_i)}{\lambda_i}} + \frac{\ln(r_i/r_{i-1})}{\frac{\ln(r_i/r_{i-1/2})}{\lambda_i} + \frac{\ln(r_{i-1/2}/r_{i-1})}{\lambda_{i-1}}} \right) \\ &\quad \left[\frac{P_{i-1,j} - P_{i,j}}{r_{i-1} - r_i} + \frac{P_{i,j} - P_{i+1,j}}{r_i - r_{i+1}} \right] \\ &= -5.1927 \times 10^{-6} m/s . \end{aligned}$$

In view of the velocity equations and the actual results from the calculation for the first grid block we, quite evidently, have a more complex calculation at hand. Of critical importance now, is the fact that the orthogonal terms no longer vanish. We see that, for example, u_{r1} is determined by the radial and angular gradients. From equation (5.1), namely,

$$\begin{bmatrix} u_r \\ u_\theta \end{bmatrix} = -\frac{1}{\mu} \begin{bmatrix} K_x \cos^2 \theta + K_y \sin^2 \theta & (K_y - K_x) \sin \theta \cos \theta \\ (K_y - K_x) \sin \theta \cos \theta & (K_x \sin^2 \theta + K_y \cos^2 \theta) \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial r} \\ 1/r \frac{\partial P}{\partial \theta} \end{bmatrix} ,$$

we can recall this dependence. Essentially, the radial component of the Darcy velocity u_r now contains a term whereby the orthogonal pressure gradient $\partial P/\partial\theta$ is forcing a fluid component in the radial direction due to the anisotropy; the higher the anisotropy, the greater the potential affect the orthogonal term can have. Intuitively, it would not be expected that the $\partial P/\partial\theta$ would have any affect on the radial term as the direction is tangent to the radial face, and hence would have no contribution to the flux term in r , as flux is flow through a given area. For this reason, we refer to the off-diagonal terms as the tangential permeability $K_t = (K_y - K_x) \sin\theta \cos\theta$.

We do, however, see the dependence of the u_r on $\partial P/\partial\theta$ in the anisotropic flow. In our case, our principal directions align with the x - and y -coordinates and our tangential permeability term, K_t , is the product of sine and cosine of the angle with respect to this coordinate system. It now means that the off-diagonal terms will still vanish if the flow is along the principal directions, but also that the K_t term is maximum at an angle of $\pi/4 = 45^\circ$ with $K_t(\pi/4) = 1/2(K_y - K_x)$. This indicates that the direction of the velocity is not necessarily orthogonal to the pressure contours as would be the case in the isotropic regime.

Radius (<i>m</i>)	θ (<i>rad</i>)	TOF (<i>sec</i>)	CTOF (<i>sec</i>)
9.0412	1.1345	0	0
7.6977	1.1257	15199	15199
6.5538	1.1205	12662	27861
5.5799	1.1093	8246.3	36107
4.7508	1.1025	6177.7	42285
4.0448	1.0906	4364.1	46649
3.4438	1.0804	3208.3	49857
2.932	1.0678	2408.8	52266
2.4963	1.0602	1683	53949
2.1254	1.0476	1246.2	55195
1.8096	1.0402	888.34	56084
1.7542	1.0472	142.73	56226
1.5407	1.0354	521.98	56748
1.4573	1.0472	194.03	56942
1.3117	1.036	304.28	57247
1.2274	1.0472	167.02	57414
1.1168	1.0378	191.96	57606
1.048	1.0472	111.92	57717
0.95085	1.0381	133.32	57851
0.80956	1.0187	188.05	58039
0.68926	0.99374	141.24	58180
0.58684	0.96092	103.3	58283
0.49963	0.97495	73.658	58357
0.42539	0.92935	51.762	58409
0.36218	0.86547	35.923	58445
0.34547	0.87266	8.8271	58454
0.30836	0.81909	19.745	58473
0.26254	0.8287	22.002	58495
0.22352	0.74573	15.886	58511
0.19031	0.73424	12.864	58524
0.16203	0.5974	8.1054	58532
0.13795	0.43855	5.1471	58537
0.11745	0.54033	6.9617	58544

Table 5.6: Anisotropic stream data. *Radius* and θ column indicate the locations of the as it passes through the grid in meters and radians, respectively. *TOF* is the time of flight of each step and the *CTOF* is the cumulative time of flight for the entire streamline measured in *seconds*.

In view of Table 5.6, we can now investigate the flow direction of the streamline

shown in Figure 5.5. From the velocity values we can see that u_{r1} and u_{r2} are negative, again, indicating that their trajectory is toward the producing well. However, u_{a1} and u_{a2} are now also negative, indicating that their respective direction is clockwise (with decreasing θ). Hence, we would expect that the angular influence to initially be toward a smaller θ value and, given the larger magnitudes, more pronounced affect on the angle. Both of these points are plainly illustrated in Figure 5.5 and is also verified by the change of θ in Table 5.6 from $1.1345rad$ to $1.1257rad$. We do note that the exit face remains the radial $r_2 = 7.9677m$ for this model where the anisotropy is an arbitrarily chosen fraction of $K_x = (0.82)K_y$.

From the TOF, we see that the first grid calculation from start $r_0 = 9.0412m$ to first face at $r_1 = 7.6977m$, a time of $15199s \approx 4.22hrs$ was calculated. As before, the radial velocity being on the order of $8 \times 10^{-5}m/s$, we find that this time would be appropriate for the distance. We also see that a larger angle is displaced given the increased angular velocity values calculated for this block. From Table 5.6, we see that roughly twice the angle is traversed with $\Delta\theta \approx 8.8 \times 10^{-3}rad$. This accounts for roughly $0.073m$ along the arc of radius $8.3m$. With the anisotropy introduced, this should be a logical result. From the Figure 5.5, the curvature of the streamline is evident.

As the exit point is carried forward and new velocities are calculated, we see the streamline data in Table 5.6 has a more pronounced change in the angular flow as compared to the isotropic case analyzed previously. Although the total variance in the angle in this case is only moderately larger at $0.0307rad$, the range of angle being from $0.4386 \leq \theta \leq 1.1345$.

And so, we see that the streamline path follows the direction expected from theory, as the flow from the outermost region flows smoothly toward the wellbore and terminating at the point $(0.11745m, 0.54033rad)$.

5.3 Future Work

The radial streamline tracing model outlined in this document offers a computationally efficient methodology to simulate the near-well flow about minute barriers near the wellbore. As fluid flows from the further extent of the reservoir towards the well, having a refining grid-structure as the flow nears the wellbore offers a greater detail in the flow path allowing the ability to include flow impedances without inferring undue burden on the computing processor.

Although this methodology is quite simple, elegant, and robust, there are omissions and limitations that would need to be addressed before such a system could be put into practice. Darcian calculations for flow in the near-well region must remain within error tolerances in instances where flow is sufficiently laminar; in general, a flow is considered laminar in cases where the Reynolds number is low, e.g. $1 < Re < 10$. The Reynolds number is given by

$$Re = \frac{\rho u D_{30}}{\mu} ,$$

where ρ is the density of the fluid, u is the Darcy velocity, D_{30} is representative of the pore size within the medium - generally taken to be 30% of the grain size - and μ the dynamic viscosity. As such, streamline modeling is not well suited to simulating flow in gas or heavy-oil scenarios.

The transformation from Cartesian to radial, presented here, allows an easy and meaningful progression from a previously existing Cartesian model to a radial model using natural advantages of the near-well geometry. Although it may be a highly simplified system, real insight into the flow regime of the region is possible with minute detail of barrier structures, say for example, in sub-surface geology or the specific well completion design. Certainly, this methodology would be an appropriate launch to

study the advantages and disadvantages of such a system for implementation in a large-scale, complex system to be modeled.

The obvious omission in the method presented in this work is that it only simulates a two-dimensional system. Although, the extension to the third dimension to include a vertical dimension would be reasonably well defined as, for the cylindrical coordinates, there is no transformation in the z -component and therefore, Pollock's method of particle tracking could be implemented verbatim. It would however, be a more complicated implementation to extend into a deviated well scenario, whereby a non-uniform grid thickness would have to be investigated if we are to follow the well trajectory. Although, a methodology for a full three-dimensional system with the ability to simulate deviated well would be a powerful addition and would be a worthwhile investigation.

The methodology as presented here, when implemented, could be a potentially effective simulator for real-time, near-well simulations. The efficiency of this model to utilize the natural physical geometry of the flow system, and the dimensionalization of the narrowing grid in the near-well region, will allow a highly detailed simulation of the near-well region, without the complication of using such high detail as we move outward. Coupling the radial grid simplification compared to traditional rectangular geometries with the inherent efficiency of streamline tracing techniques.

A highly efficient simulation tool would be an especially useful tool during well-testing. Many well-tests involve altering the simulated flow rates and investigating the effects on bottom-hole pressure. Given the simplicity of this model, it may be possible to adapt and run a streamline simulation in near real-time if the physical reservoir data grid is established and determinable.

In discretizing the mathematics presented here, it would be useful to compare the two-point flux approximation (TPFA) with more intricate numerical approximations.

The TPFA works well and efficient in systems of Cartesian coordinates where grids are of regular and uniform dimensions. However, it has been shown that it can introduce numerical errors into the streamline calculations [Aarnes *et al*, 2007]. It may be more accurate to incorporate multi-point (MPFA) or corner-point flux approximations (CPFA) as they are generally regarded as being more accurate as there is more pressure data accounted for equivalent grid geometries. See [Hægland, 2009] for a concise discussion on various geometries and discretization techniques.

5.4 Concluding remarks

Streamline simulation techniques have the potential of being an extremely useful and powerful means of flow visualization in the near-well region. With the geometry as laid out in this work, we have the potential to simulate how known well-completion barriers and geological structure affect fluid flow, as well as a multitude of other reservoir data, will affect the path of fluid from/to a well. Generally speaking, in reservoir characterization, most if not all of the hard data we have on geological structure within the reservoir is collected from drilling operations, and to a large extent, typical reservoir simulators cannot incorporate this data in full. By this technique, the potential to incorporate multitudes of information and local heterogeneities into the simulation could be significantly improved.

This methodology has the potential to be a useful tool for the simulation of the near-well region of a reservoir. Currently, there exists no commercially available streamline simulator for this type of usage. It would be a useful matter to expand this methodology to three-dimensions and a comprehensive comparative analysis of this method with standard finite difference mathematical models.

This work stands as a proof-of-concept for a streamline model of the near-well

region with the potential to be tremendously viable as a completion tool. However, for this model to be full-functioning, further research to extend the model will be required.

Appendix A

TPFA for Pressure Calculation

The code for the flux approximation is contained herein. To make things easier to code, the expression in equation (4.6) is broken into segments and assembled at the end. As shown in the box below, the TPFA calculation is separated into three parts – the permeability matrix, the transmissibility matrix, and the TPFA discretization – and the information is combined at the end to solve the system and calculate the fluxes. All code is written by the author.

```
function [P,V] = TPFA(Grid, K, q)
% Set up grid and permeability matrix.
\\
Nx = Grid.Nx; Ny = Grid.Ny; \\
Nz = Grid.Nz; N = Nx{*}Ny{*}Nz; \\
hx = Grid.hx; hy = Grid.hy; \\
hz = Grid.hz; L = K.(-1);\\
% Build transmissibility matrix
\\
tx = 2{*}hy{*}hz/hx; \\
```

```

TX = zeros( Nx+1, Ny, Nz); \\
ty = 2{*}hx{*}hz/hy; \\
TY = zeros( Nx, Ny+1, Nz); \\
tz = 2{*}hx{*}hy/hz; \\
TZ = zeros( Nx, Ny, Nz+1);\\
TX( 2:Nx, :, :) = tx./(L (1,1:Nx-1, :, :)+L(1,2:Nx, :, :));\\
  TY( :, 2:Ny, :) = ty./(L (2, :, 1:Ny-1, :)+L(2, :, 2:Ny, :)); \\
TZ( :, :, 2:Nz) = tz./(L (3, :, :, 1:Nz-1)+L(3, :, :, 2:Nz));\\
% Assemble TPFA matrix.
\\
x1 = reshape (TX(1:Nx, :, :), N, 1); \\
x2 = reshape (TX (2:Nx+1, :, :), N, 1); \\
y1 = reshape (TY(:, 1:Ny, :), N, 1);\\
  y2 = reshape (TY ( :, 2:Ny+1, :), N, 1); \\
z1 = reshape (TZ(:, :, 1:Nz), N, 1); \\
z2 = reshape (TZ ( :, :, 2:Nz+1), N, 1);\\
DiagVecs = {[ ]-z2, -y2, -x2, x1+x2+y1+y2+z1+z2, -x1, -y1, -z1[ ]}; \\
DiagIndx = {[ ]-Nx{*}Ny, -Nx, -1, 0, 1, Nx, Nx{*}Ny[ ]};\\
A = spdiags( DiagVecs, DiagIndx, N, N); \\
A(1,1) = A(1,1) + sum(Grid.K(:, 1, 1, 1));\\
% Solve the linear system and extract interface fluxes.
\\
u = A\q; \\
P = reshape (u, Nx, Ny, Nz);\\
V.x = zeros (Nx+1, Ny, Nz); \\
V.y = zeros (Nx, Ny+1, Nz); \\

```

```
V.z = zeros (Nx, Ny, Nz+1); \\
V.x(2:Nx, :, :) = (P(1:Nx-1, :, :) - P(2:Nx, :, :)).{*}TX(2:Nx, :, :); \\
V.y(:, 2:Ny, :) = (P(:, 1:Ny-1, :) - P(:, 2:Ny, :)).{*}TY(:, 2:Ny, :); \\
V.z(:, :, 2:Nz) = (P(:, :, 1:Nz-1) - P(:, :, 2:Nz)).{*}TZ(:, :, 2:Nz); \\
```

Appendix B

Radial Grid Setup Matlab Code

This code employs the TPFA code from Appendix A to solve for the pressure distribution that will be required for the impending streamline simulation. This code sets up all the variables required by the TPFA code - the Cartesian grid dimensions, permeability information, etc - and extracts the pressure data as required for the radial geometry. Whether the permeability field is simulated as isotropic or anisotropic in xy -plane can be set within this code prior to running. All code written by the author.

Contents

- To set the Grid size in each direction.
- To set the permeability grid.
- no flow block.
- Anisotropy in xy -direction.
- no flow in z -direction.
- Grid size.

- The flow is described by:
- The pressure is then invoked by the command
- Plot of pressure contours
- Angular and Radial grid-steps defined
- Scaling grid to overlay on pressure contours
- Define the node locations.
- Extracting the pressure value at each node.
- Plot node locations and overlay pressure contour in subplot
- Reshape and organize radial pressure values
- Plot the radial pressure and arrange in radial form in subplot
- Plot radial pressure values on square grid directly to view data

To set the Grid size in each direction.

```
Grid.Nx = 501; Grid.hx = 1/Grid.Nx; Grid.Ny = 501;
Grid.hy = 1/Grid.Ny; Grid.Nz = 1; Grid.hz = 1/Grid.Nz;
```

To set the permeability grid.

```
Grid.K = 0.5.*ones(3,Grid.Nx, Grid.Ny).*10(-9);
%10^(-9) chosen to be 1 Darcy(^-12 m^2)/ cp viscosity(^-3 Pa s)
```

```
% Random perm values
```

```
%Grid.K = exp(smooth3(smooth3(randn(3,Grid.Nx, Grid.Ny))));
```

no flow block.

```
%Grid.K(1:2,100:300,200:300) = 0.002*10(-9);
```

Anisotropy in xy-direction.

```
Grid.K(1,::) = Grid.K(1,::){*}.82; Grid.K(2,::)
= Grid.K(2,::); %Grid.K(1,
```

no flow in z-direction.

```
Grid.K(3,::) = 0;
```

Grid size.

```
N = Grid.Nx{*} Grid.Ny{*} Grid.Nz;
```

The flow is described by:

```
q = zeros(N,1); q(1) = {[]-1{*}10(-3){]}; q(N) =
{[]1{*}10(-3){]};
```

The pressure is then invoked by the command

```
{[]P,V{]} = TPFA (Grid, Grid.K,q);
```

Plot of pressure contours

```
figure contourf(P,28) axis square axis off % hold on
```

```
% mesh(P)

% figure
\\
% quiver(V.x', V.y')

%The following two plots convert the polar data to cartesian so that we can
\\
%plot out the info and verify that it is doing what its supposed to!
\\
```



Angular and Radial grid-steps defined

The next section computes the circular radii given the start 'r0' and the angle between rays ('Delta').

```
Delta = pi/18; {[r]} = 0.1; rstep = {[r]};
while rstep < 10 {[rstep]} = r(1,length(r)) * Delta
+ r(1, length(r)); {[r]} = {[r, rstep]}; end

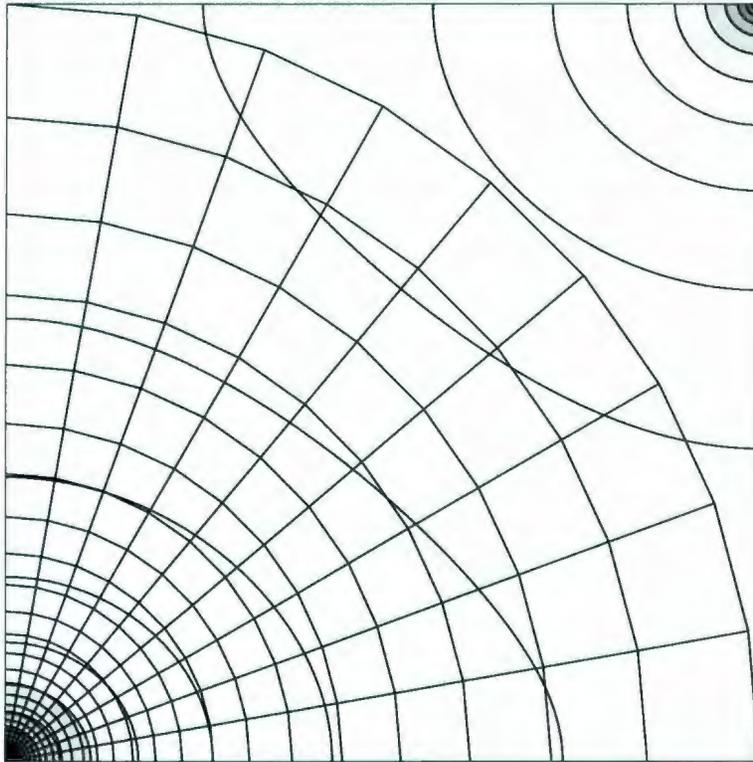
clear rstep
```

Scaling grid to overlay on pressure contours

```
%Redefine the radial steps to be scaled to the number of grid-blocks
% in the simulation.
\\
scale = (length(P)-1) / max(r); \\
rscale = scale*[r];

theta = 0:Delta:pi/2; \\
hold on\\
{[X, Y]} = pol2cart(meshgrid(theta, rscale), (meshgrid(rscale,theta)'));

X = X +ones(size(X)); \\
Y = Y +ones(size(Y)); \\
line(X,Y,'Color','k') line(X',Y', 'Color','k') \\
axis square \\
axis off
```



Define the node locations.

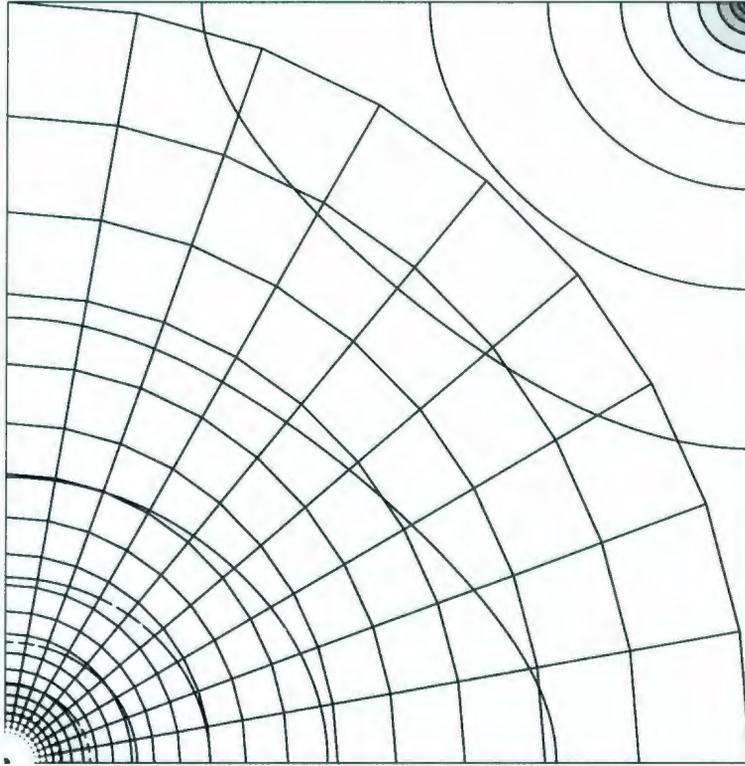
```
%First we define the angles...
theta2 = pi/36:Delta:pi/2; %Then we define the node radii...

% Logarithmic mean for the radial node locations within the blocks.

nodeR = {}; \
for i = 1:(length(r)-1) rn = (r(i+1) - r(i))/(log(r(i+1)/r(i))); \
    {}nodeR{} = {}nodeR, rn{}; \
end
```

```
nodelocX = {[]{[]}; \\
nodelocY = {[]{[]}; \\
for i =1:length(theta2) \\
phi = theta2(1,i){*} ones(max(size(nodeR),1));\\
    {[]nodeX,nodeY{[]}} = pol2cart(phi,nodeR); \\
{[]nodelocX{[]}} = {[]nodelocX;nodeX'{}; \\
{[]nodelocY{[]}} = {[]nodelocY;nodeY'{}; \\
end

% to move the nodes so as they plot out correctly...
\\
addX = ones(size(nodelocX)); \\
addY = ones(size(nodelocY)); \\
nodelocXA = scale.*{[]}nodelocX + addX; \\
nodelocYA = scale.*{[]}nodelocY + addY; \\
plot(nodelocXA,nodelocYA,'w.')
```



Extracting the pressure value at each node.

```

point = round({[]nodelocXA, nodelocYA{[]}); Pold
= {[]-[]}; Prad = {[]-[]}; Krad = {[]-[]};\
%K_old = [];
\
K.x = reshape(Grid.K(1,:,:), Grid.Nx, Grid.Ny); \
K.y = reshape(Grid.K(2,:,:), Grid.Nx, Grid.Ny);

for i = 1:length(point) Pold = Prad;\
%K_old = K_rad;
\
Prad = P(point(i,1),point(i,2)); \

```

```
Krad(1,i) = K.x(point(i,1),point(i,2)); Krad(2,i) = K.y(point(i,1),point(i,2));  
\\  
Prad = {[]Pold, Prad{]}; \  
%K_rad = [K_old,K_rad];  
  
end clear Pold
```

Plot node locations and overlay pressure contour in subplot

```
%figure  
%subplot(1,3,1)  
%contourf(P,25)  
%axis square  
%hold on  
%plot(node_locXA, node_locYA,'b.')
```

Reshape and organize radial pressure values

```
PradY = reshape(nodelocXA,29,9); \  
PradX = reshape(nodelocYA, 29,9); \  
Prad = reshape(Prad, 29,9);\  
% P_rad is now organized.  
% P(i,j) yields i steps in radial (from out to in), j in angular (theta 0->90).  
%To visually verify that this is correct, show the following meshplot.  
Krad = reshape(Krad,2,29,9);
```

Plot the radial pressure and arrange in radial form in subplot

```
%hold on
%subplot(1,3,2)
%pcolor(P_rad) % See P_rad matrix arrangment
%pcolor(P_radX,P_radY,P_rad)% display radial pressures
%shading interp
%axis off
```

Plot radial pressure values on square grid directly to view data

```
%axis square
%subplot(1,3,3);
%pcolor(P_rad);
%shading interp
%axis square
```

Appendix C

Streamline Simulation for Radial Grid Code

To run this code, the *RadialGrid* code in Appendix B must be run a-priori as the variables created from that code are required as input for the streamline simulation herein. All code written by the author.

Contents

- clear variables
- Set up the grid dimensions
- To plot the grid output, include these lines:
- Organize the conductivity matrix appropriately.
- Data for streamline start points
- For Loop to count through start positions
- Find matrix block to start at

- Streamline loop.
- Determine pt.new in grid
- Begin the streamline calculations
- Face Discharges & Upscaling
- Velocity Calc
- Area of grid-cell faces
- Travel-time
- least time determination
- prepare pt.old for line
- exit location
- End of while loop

```
function [stream, tof, ctof] = SLR(r,theta,node_R,K_rad,P_rad)
```

```
clear variables
```

```
clear N P V q nodelocX nodelocXA nodelocY nodelocYA
```

```
clear addX addY X Y i rn
```

```
%RadialSetup
```

Set up the grid dimensions

```

rad = fliplr(r); \\  

ang = fliplr(theta); \\  

grid.r = meshgrid(rad, ang)'; \\  

grid.ang = meshgrid(ang, rad); \\  

%P_rado = P_rad;  

Prad = flipud(Prad); \\  

clear rad ang  
  

{[]idim, jdim[]} = size(Prad);

```

pause on

To plot the grid output, include these lines:

```

{[]X,Y[]} = pol2cart(grid.ang, grid.r); \\  

figure hold on line(X, Y, 'Color', 'k') line (X', Y', 'Color', 'k') axis  

square

```

```

rad = fliplr(nodeR); \\  

ang = fliplr(pi/36 : mean(diff(theta)) : \\  

pi/2); node.r = meshgrid(rad, ang)'; \\  

node.ang = meshgrid(ang,rad); clear rad ang

```

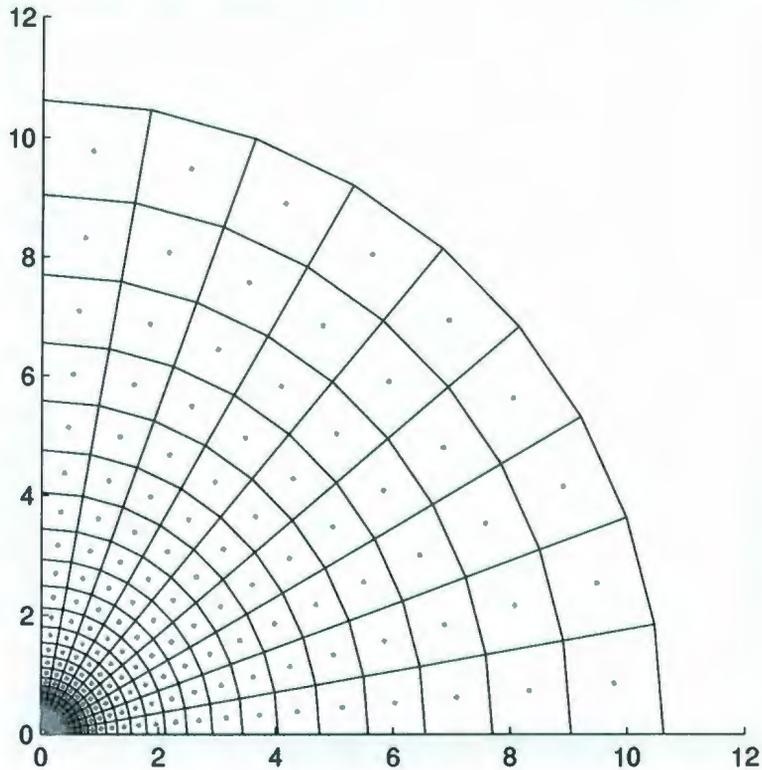
```

{[]X,Y[]} = pol2cart(node.ang, node.r); hold on plot(X, Y, '.r') axis  

square

```

```
{[]stream[]} = {[]{[]}};
```



Organize the conductivity matrix appropriately.

```
Krad.x = reshape(Krad(1,:,:), 29, 9); \\
Krad.y = reshape(Krad(2,:,:), 29, 9); \\
Krad.a = (Krad.x .* (cos(node.ang)) . 2) + \\
(Krad.y .* (sin(node.ang)) . 2); \\
Krad.b = (Krad.y - Krad.x) .* (sin(node.ang) .* cos(node.ang)); \\
Krad.c = ((Krad.x .* (sin(node.ang)) . 2)... \\
+ (Krad.y .* (cos(node.ang))) . 2) .* ((1./ (node.r).2)) ; \\
Krad.z = {[]{[]}}; \\
%clear K_rad
\\
```

Data for streamline start points

data struct give us our starting points for the streamlines.

The outside edge of the grid at the second cylindrical row at equal angle to the nodes is chosen.

```

%data.r = grid.r(2) .* ones(1,9);
%data.ang = node.ang(2,:);

% to plot the data points:
%           [data.x, data.y] = pol2cart(data.ang, data.r);
%           plot(data.x, data.y, 'bx')

% Start Locations
% Chosen to start on the outside edge of the
start = [grid.r(2, 1:(end-1)) ; node.ang(1,:)]; %start = [
[mpt, npt] = size(start);

pt.new = {}; pt.old = {};

```

For Loop to count through start positions

```

%for n=2:7

```

```
%count =
```

Find matrix block to start at

```
{[]pt.new{[]} = {[]start(2,3) , start(1,3){[]};\
```

```
%[pt.new] = [start(2, 3) , start(2,3)];
```

```
\
```

Streamline loop.

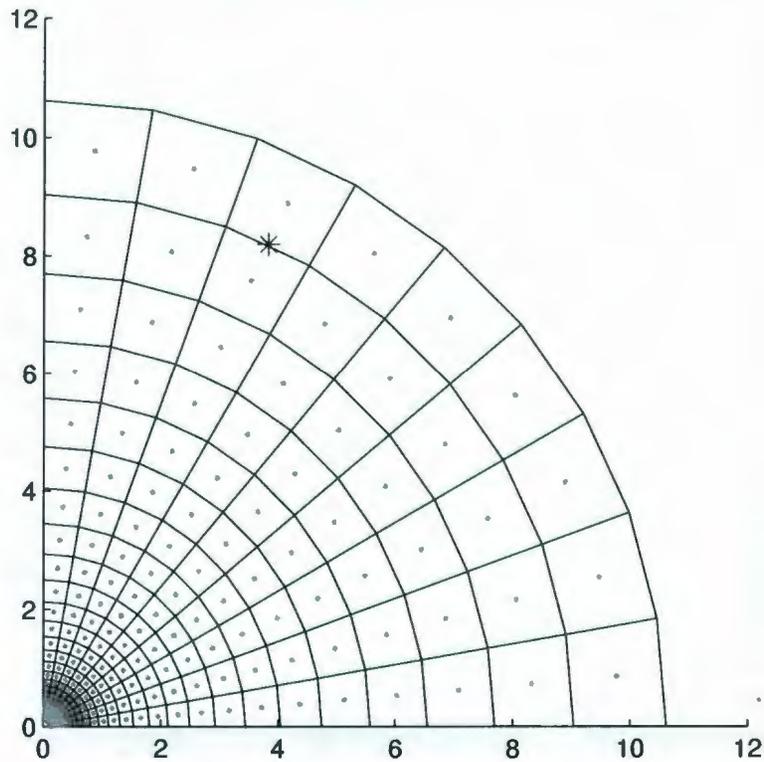
```
stream.ang = {[]{[]}; stream.r = {[]{[]};
```

```
pt.old = pt.new;
```

```
stream = {[]pt.old{[]};
```

```
{[]pt.newx, pt.newy{[]} = pol2cart(pt.new(1), pt.new(2)); plot(pt.newx,  
pt.newy, 'b{*}',... 'MarkerSize', 9)
```

```
C={[]{[]}; ctof = {[]{[]}; tof={[]{[]};
```



```
while stream(end,2) \textgreater{}= grid.r(end-1)
```

Determine pt.new in grid

```
----->>>>>>>
```

```
%-----H E R E-----%%
```

```
{[]A[]} = find(stream(end,2) \textless{}= grid.r(:,1)); \\  
{[]B[]} = find(stream(end,1) \textless{}= grid.ang(1,:));
```

```
{[]A[]} = A(end); {[]B[]} = B(end);
```



```
A1=A; B1=B;
```

Face Discharges & Upscaling

To control the boundaries, we use the following if statements

```

    if A == 1, break; \\
else A1 = A; \\
end if A == idim, break; \\
else A1 = A; \\
end if B == 1, break; \\
else B1 = B; \\
end if B == jdim, break; \\
else B1 = B; \\
end

```

Velocity Calc

face velocity of 1st radial face

```

\\
P.r1 = ( Prad(A,B) - Prad(A1-1,B)) ./ (node.r(A,B) - node.r(A1-1,B));
\\
K.r1 = ( log(node.r(A1-1,B)/node.r(A,B)) ) ./ ... \\
( (log(node.r(A1-1,B)/grid.r(A,B)) ./ (Krad.a(A1-1,B))) +... \\
(log(grid.r(A,B)/node.r(A,B)) ./ Krad.a(A,B)) ); \\
%P.ra1 = (Prad(A1-1,B-1)-Prad(A1-1,B+1))./(node.ang(A1-1,B-1)-node.ang(A1-1,B+1)
\\

```

```

P.ra1 = 0.5.*{(Prad(A,B+1)-Prad(A,B))./(node.ang(A,B+1)-node.ang(A,B))
+ ... \\
(Prad(A,B)-Prad(A,B-1))./(node.ang(A,B)-node.ang(A,B-1))};\\
%      K.ra1 = (Krad.b(A1-1,B1-1) + Krad.b(A1-1, B1+1)) ./ ...
\\
%      (node.ang(A1-1, B1-1) - node.ang(A1-1, B1+1));
\\
%Harmonic mean of three blocks.
\\
K.ra1 = 3./((1/Krad.b(A,B-1))+(1/Krad.b(A,B))+(1/Krad.b(A,B+1)));
\\
u.r1 = -(K.r1{*}P.r1 + K.ra1{*}P.ra1/grid.r(A,B));\\
% face velocity of 2nd radial face
\\
P.r2 = ( Prad(A1+1,B) - Prad(A,B)) ./ (node.r(A1+1,B) - node.r(A,B));\\
K.r2 = ( log(node.r(A,B)/node.r(A1+1,B)) ) ./ ... \\
( (log(node.r(A,B)/grid.r(A1+1,B)) ./ (Krad.a(A,B))) +... \\
(log(grid.r(A1+1,B)/node.r(A1+1,B)) ./ Krad.a(A1+1,B)) );\\
%P.ra2 = (Prad(A,B1-1)-Prad(A,B1+1))./ ...
\\
% (node.ang(A,B1-1)-node.ang(A,B1+1));
\\
P.ra2 = 0.5.*{(Prad(A,B+1)-Prad(A,B))./(node.ang(A,B+1)-node.ang(A,B))
+ ... \\
(Prad(A,B)-Prad(A,B-1))./(node.ang(A,B)-node.ang(A,B-1))}; \\
%K.ra2 = (Krad.b(A,B1-1) + Krad.b(A, B1+1)) ./ ...

```

```

\\
% (node.ang(A, B1-1) - node.ang(A, B1+1));
\\
%Harmonic mean of three blocks
\\
K.ra2 = 3./((1/Krad.b(A,B-1))+(1/Krad.b(A,B))+(1/Krad.b(A,B+1)));
\\
q.r1 = K.r1{*}P.r1 + K.ra1{*}P.ra1/grid.r(A,B);\\
u.r2 = -(K.r2{*}P.r2 + K.ra2{*}P.ra2./grid.r(A1+1,B));\\
%face velocity of 1st angular face
\\
P.a1 = ( Prad(A,B) - Prad(A, B1-1)) ./ (node.ang(A,B) - node.ang(A,
B1-1)); \\
K.a1 = (Krad.c(A,B) + Krad.c(A, B1-1)) ./ (2);\\
%P.ar1 = ( Prad(A1+1,B) - Prad(A1-1,B)) ./ ...
\\
%(node.r(A1+1,B) - node.r(A1-1,B));
\\
P.ar1 = 0.5.*{( Prad(A+1,B)-Prad(A,B))./(node.r(A+1,B)-node.r(A,B))+...
\\
(Prad(A,B)-Prad(A-1,B))./(node.r(A,B)-node.r(A-1,B)) }; \\
K.ar1 = ( log(node.r(A1-1, B) ./ node.r(A1+1, B) ) ) ./ (... \\
(log(node.r(A1-1,B)./grid.r(A,B))./Krad.b(A1-1,B)) +... \\
(log(grid.r(A,B)./grid.r(A1+1,B))./Krad.b(A,B)) +... \\
(log(grid.r(A1+1,B)./node.r(A1+1,B))./Krad.b(A1+1,B)) ) ;\\
u.a1 = -(K.ar1{*}P.ar1 + K.a1{*}P.a1/grid.ang(A,B));\\

```

```

%face velocity of 2nd angular face
\\
P.a2 = ( Prad(A,B1+1) - Prad(A, B)) ./ (node.ang(A,B1+1) - node.ang(A,
B)); \\
K.a2 = (Krad.c(A,B1+1) + Krad.c(A, B)) ./ (2);\\
%P.ar2 = ( Prad(A1+1,B) - Prad(A1-1,B)) ./ ...
\\
%(node.r(A1+1,B) - node.r(A1-1,B));
\\
P.ar2 = 0.5.*(( Prad(A+1,B)-Prad(A,B))./(node.r(A+1,B)-node.r(A,B))+...
\\
(Prad(A,B)-Prad(A-1,B))./(node.r(A,B)-node.r(A-1,B)) ); \\
K.ar2 = ( log(node.r(A1-1, B) ./ node.r(A1+1, B) ) )./ (...\\
(log(node.r(A1-1,B)./grid.r(A,B))./Krad.b(A1-1,B)) +...\\
(log(grid.r(A,B)./grid.r(A1+1,B))./Krad.b(A,B)) +... \\
(log(grid.r(A1+1,B)./node.r(A1+1,B))./Krad.b(A1+1,B)) ); \\
u.a2 = -(K.ar2.*P.ar2 + K.a2.*P.a2/grid.ang(A,B));

% change the theta partial of P to stay in line with the central node for
\\
% the off-diagonal portions of the equations. Yields higher fluxes and
\\
% velocities in the r direction.
\\
%q
\\

```

Area of grid-cell faces

We note here that the calculations below have $\Delta(z)=1$, and are therefore not included in the cross-sectional area multiplication

```

%u.r1 = q.r1/(1*grid.r(A,B).*(grid.ang(A,B1+1)-grid.ang(A,B)));
%u.r1 = -q.r1/(1*grid.r(A,B).*(Del_theta));

%u.r2 = q.r2/(1*grid.r(A1+1,B).*(grid.ang(A,B1+1)-grid.ang(A,B)));
%u.r2 = -q.r2/(1*grid.r(A1+1,B).*(Del_theta));

%u.a1 = -q.a1/(1*abs(grid.r(A1+1,B)-grid.r(A,B)));

%u.a2 = -q.a2/(1*abs(grid.r(A1+1,B)-grid.r(A,B)));

u\

    r1=grid.r(A,B) \
    r2=grid.r(A+1,B) \
    a1 = grid.ang(A,B) \
    a2 = grid.ang(A,B+1)

r.e=pt.new(2); \

```

```
ang.e=pt.new(1);
```

```
Ar=r1.*r2.*(u.r2-u.r1)/(r1-r2);
```

```
\\
```

```
b = u.r1 - Ar./r1;
```

```
Aa = (u.a2-u.a1)/(a2-a1); \\
```

```
ba = u.a1-(Aa.*a1);
```

Travel-time

```
t.r2 = ((r2-r.e)/(b))-(Ar/(b2)).*log((Ar+b.*r2)/(Ar+b.*r.e));
```

```
t.r1 = (r1-r.e)/(b)-(Ar/(b2)).*log((Ar+b.*r1)/(Ar+b.*r.e));
```

```
t.a1 = (1./Aa).*log((Aa.*ang.e+ba)/(Aa.*a1+ba));
```

```
t.a2 = (1./Aa).*log((Aa.*ang.e+ba)/(Aa.*a2+ba));
```

least time determination

```
t.x = {[]t.r1,t.r2,t.a1,t.a2{]}; t.p = find(t.x\textgreater{0});
```

```
t.x = min(t.x(t.p))
```

```
t
```

prepare pt.old for line

```
{[]pt.oldx,pt.oldy{]}=pol2cart(stream(end,1),stream(end,2));
```

exit location

```

if t.x == t.r2

r.x = r2; \\
%ang.x = ((1/Aa)*( (exp(t.x*Aa))*(Aa*pt.new(1)+ba))-ba/Aa);
\\
ang.x = ang.e+((Aa{*}ang.e+ba)/r.e){*}t.x-0.5{*}((u.a1-u.a2)/r.e){*}t.x;\\
disp('t.x = t.r2')

{[]pt.newx, pt.newy{[]}} = pol2cart(ang.x, r.x); plot(pt.newx, pt.newy,
'b{*}',... '\\
MarkerSize', 9) line({[]pt.oldx,pt.newx{[]}},{[]pt.oldy,pt.newy{[]}})

elseif t.x == t.r1 r.x = r1; \\
%ang.x = (1/Aa)*(( (exp(t.x*Aa)).*(Aa*pt.new(1)+ba))-ba);
\\
ang.x = ang.e+((Aa{*}ang.e+ba)/r.e){*}t.x-0.5{*}((u.a1-u.a2)/r.e){*}t.x;
\\
disp('t.x = t.r1')

{[]pt.newx, pt.newy{[]}} = pol2cart(ang.x, r.x);\\
plot(pt.newx, pt.newy, 'b{*}',... '\\
MarkerSize', 9) line({[]pt.oldx,pt.newx{[]}},{[]pt.oldy,pt.newy{[]}})

elseif t.x == t.a1

```

```

disp('t.x = t.a1') ang.x=a1; \\  

r.n = node.r(A,B); u.e=Ar./(r.e)+b; \\  

u.n=Ar./(r.n)+b; \\  

%r.x = r.e+u.e.*t.x;  

\\  

r.x = r.e+u.e.*t.x-0.5.*(Ar./(r.n2)).*u.n*(t.x2);  
  

[pt.newx, pt.newy] = pol2cart(ang.x, r.x); \\  

plot(pt.newx, pt.newy, 'b*',... '\\  

MarkerSize', 9) line([pt.oldx,pt.newx],[pt.oldy,pt.newy])  
  

elseif t.x == t.a2 disp('t.x = t.a2') ang.x = a2;\\  

r.n = node.r(A,B); \\  

u.e=Ar./(r.e)+b; \\  

u.n=Ar./(r.n)+b; \\  

%r.x = r.e+u.e.*t.x;  

\\  

r.x = r.e+u.e.*t.x-0.5.*(Ar./(r.n2)).*u.n*(t.x2);  
  

% r.x = r.e+u.e.*t.x-0.5.*(Ar./(r.e^2)).*u.e*(t.x^2);  
  

[pt.newx, pt.newy] = pol2cart(ang.x, r.x); \\  

plot(pt.newx, pt.newy, 'b*',... 'MarkerSize', 9)  
  

line([pt.oldx,pt.newx],[pt.oldy,pt.newy]) \\  


```

```
else display('something went wrong - line 225-268...likely the time  
calc') \\
```

```
break
```

```
end
```

```
stream = {[ ]stream;ang.x, r.x{ ]}
```

```
tof = {[ ]tof; t.x{ ]};
```

```
ctof = {[ ]ctof; sum(tof){ ]};
```

```
pt.new = {[ ]stream(end,1),stream(end,2){ ]};
```

End of while loop

```
end
```

Bibliography

- [Arfken, 1985] **Arfken, G. (1985)**. *Mathematical Methods For Physicists* (3rd Edition), Academic Press Inc.
- [Aarnes *et al*, 2007] **Aarnes, J.E., Gimse, T., Lie, K.-A. (2007)**. *An introduction to the numerics of flow in porous media using Matlab*, in Geometrical Modeling, Numerical Simulation, and Optimization: Industrial Mathematics at SINTEF, Springer Verlag.
- [Aziz & Settari, 1979] **Aziz, S. & Settari, A. (1979)**. *Petroleum Reservoir Simulation*, Applied Science Publishers.
- [Batycky *et al*, 1996] **Batycky, R.P., Blunt, M.J., and Thiele, M.R. (1997)**. *A 3D Field-Scale Streamline-Based Reservoir Simulator*, SPERE 12 (4): 246-254. SPE-36726-PA. doi: 10.2118/36726-PA.
- [Bear, 1972] **Bear, J. (1972)**. *Dynamics of Fluids in Porous Media*, Dover.
- [Bratvedt *et al*, 1993] **Bratvedt, F., Bratvedt, K., Buchholz, C.F., Gimse, T., Holden, H.L., Risebro, N.H. (1993)**. *Frontline and frontsim: two full scale, two-phase, black oil reservoir simulators based on front tracking*, Surveys on Mathematics for Industry, Vol. 3, pp. 185-215.

- [Bratvedt *et al*, 1996] **Bratvedt, F., Gimse, T., Tegnander, C. (1996)**. *Streamline computations for porous media flow including gravity*, Transport in Porous Media, Vol. **25**(1), pp. 63-78.
- [Carslaw & Jaeger, 1959] **Carslaw, H.S. & Jaeger, J.C. (1959)**. *Conduction of Heat in Solids*, Second Edition. Oxford University Press.
- [Cheng *et al*, 2005] **Cheng, H., Osaka, I., Datta-Gupta, A., King, M.J. (2005)**. *A Rigorous Compressible Streamline Formulation for Two- and Three-Phase Black-Oil Simulation*. SPEJ, Vol. **11**(4), pp. 407-417. SPE-96866-PA.
- [Datta-Gupta & King, 2007] **Datta-Gupta, A. and King, M.J. (2007)**. *Streamline Simulation: Theory and Practice*, Textbook Series, SPE, Richardson, Texas 11.
- [Datta-Gupta *et al*, 2001] **Datta-Gupta, A., Kulkarni, K. N., Yoon, S., Vasco, D. W. (2001)**. *Streamlines, ray tracing, and production tomography: generalization to compressible flow*, Petroleum Geoscience, Vol. 7 2001, pp. S75-S86.
- [Donea & Huerta, 2003] **Donea, J. & Huerta, A. (2003)**. *Finite Element Methods for Flow Problems*, Wiley & Sons Ltd.
- [Fi1Kaiv8, 2010] **Fi1Kaiv8, Wikipedia User. (2010)**. *Streamlines, streaklines, and pathlines*, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, Inc. Retrieved August 30, 2012, from en.wikipedia.org/wiki/File:Streaklines_and_pathlines_animation.gif
- [Freeze & Cherry, 1979] **Freeze, R.A., Cherry, J.A. (1979)**. *Groundwater*, Prentice-Hall.

- [Hægland, 2009] **Hægland, H. (2009)**. *Streamline methods with application to flow and transport in fractured media*, Ph.D. Thesis, University of Bergen, Dept. of Mathematics.
- [Hægland *et al*, 2006] **Hægland, H., Dahle, H.K., Eigestad, G.T., Lie, K.-A., Aavatsmark, I. (2006)**. *Improved streamlines and time-of-flight for streamline simulation on irregular grids*, *Advances in Water Resources*, Vol. **30**(4), pp. 1027–1045.
- [Huang *et al*, 2004] **Huang, W., Di Donato, G., Blunt, M.J. (2004)**. *Comparison of streamline-based and grid-based dual porosity simulation*, *J of Petro Science and Eng*, 43, pp. 129-37.
- [King & Datta-Gupta, 1998] **King, M.J., Datta-Gupta, A. (1998)**. *Streamline Simulation: A Current Perspective*, *In Situ*, Vol 22, 1, pp. 91-140.
- [Logan, 2002] **Logan, D. L. (2002)**. *A First Course in the Finite Element Method, Third Edition*, Brooks/Cole.
- [McDonald & Harbough, 1988] **McDonald, M.G., Harbough, A.W. (1988)**. *A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model*, *Techniques of Water-Resources Investigations of the United States Geological Survey*, Book 6, Chapter A1, pp. 2-1 - 3- 28.
- [Mei, 1995] **Mei, C. (1995)**. *Mathematical Analysis in Engineering*, Cambridge University Press.
- [Morse & Feshbach, 1953] **Morse, P.M. & Feshbach, H. (1953)**. *Methods of Theoretical Physics*, McGraw-Hill, New York.

- [Muskat & Wyckoff, 1934] **Muskat, M. & Wyckoff, R.D. (1934)**. *A Theoretical Analysis of Water-Flooding Networks*, Transactions of the AIME, Vol. **107**(1), pp. 62-77.
- [Peaceman, 1978] **Peaceman, D.W. (1978)**. *Interpretation of Well-Block Pressures in Numerical Reservoir Simulation*, Soci. of Petroleum Eng. Journal, Paper Num. 6893-PA; June 1978, pp. 183-94.
- [Peaceman, 1983] **Peaceman, D.W. (1983)**. *Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability*, SPEJ, Paper Num. 10528, ; June 1983, pp. 531-43.
- [Pollock, 1988] **Pollock, D.W. (1988)**. *Semianalytical Computation of Path Lines for Finite-Difference Models*, Ground Water; Vol. **26**(6), pp. 743-50.
- [Prévost *et al*, 2002] **Prévost, M., Edwards, M.G., Blunt, M.J. (June 2002)**. *Streamline tracing on curvi-linear structured and unstructured grids*, SPE Journal, pages 139–148.
- [Saatdjian, 2000] **Saatdjian, E (2000)**. *Transport Phenomena: Equations and Numerical Solutions*, Wiley & Sons Ltd.
- [Schwabe & Brand, 1967] **Schwabe, K. & Brand, J. (1967)**. *Prediction of Reservoir Behaviour Using Numerical Simulators*, Soci. of Petroleum Engr., 42nd Annual Fall Meeting Paper Num. 1857.
- [Skinner & Johansen, 2011] **Skinner, J.H., Johansen, T.E. (2011)**. *Near Wellbore Streamline Modeling: Its Novelty, Application and Potential Use*, Soci. of Petroleum Engr., SPE Int'l Symposium and Exhibition on Formation Damage Control, SPE 149962-PP.

- [Thiele, 2001] **Thiele, M. (2001)**. *Streamline Simulation*, 6th Int'l Forum on Reservoir Simulation, Sept 3-7, 2001. Schloss Fuschl, Austria.
- [Thiele, 2003] **Thiele, M. (2003)**. *Streamline Simulation*, 7th Int'l Forum on Reservoir Simulation, June 23rd - 27th, 2003. Baden-Baden, Germany.
- [Vanfai *et al*, 2005] **Vanfai, K. (2005)**. *Handbook of Porous Media*, CRC Press.
- [van Poolen *et al*, 1970] **van Poolen, H.K. & Jargon, J.R. (1970)**. *Individual Well Pressures in Reservoir Modeling*. Oil and Gas J.; October 1970, pp. 78-80.
- [Zaslavsky, 1962] **Zaslavsky, D. (1962)**. *Flow model of water in saturated soil with non-uniform conductivity*. Dept. of Civil Engineering, Technion-I.I.T., Haifa, Pub. 15.
- [Zheng & Bennett, 2002] **Zheng, C. & Bennett, G.D. (2002)**. *Applied Contaminant Transport Modeling (2nd Edition)*, Wiley-Interscience. LIB ID: TD 426 Z43 2002
- [Zijl, 1996] **Zijl, W. (1996)**. *The Symmetry Approximation for Nonsymmetric Permeability Tensors and its Consequences for Mass Transport*, Transport in Porous Media, Vol. 22, pp. 121-136, Kluwer Academic Publishers.

