2014-01-01

A Circuit-Based Algorithm For Two-Phase Flow At The Pore Scale In Porous Media

Todd Dorethy
University of Texas at El Paso, twdorethy@miners.utep.edu

A CIRCUIT-BASED ALGORITHM FOR TWO-PHASE FLOW AT THE PORE SCALE IN POROUS MEDIA

TODD WAYNE DORETHY JR.

Computational Science

APPROVED:

Vinod Kumar, Ph.D., Chair

Son-Young Yi, Ph.D.

Pavana Prabhakar, Ph.D.

Kiran Katta, Ph.D.

Bess Sirmon-Taylor, Ph.D.
Interim Dean of the Graduate School

Dedication

To Mom, Dad, Joshua and Nathan

A CIRCUIT-BASED ALGORITHM FOR TWO-PHASE FLOW AT THE PORE SCALE IN POROUS MEDIA

by

TODD WAYNE DORETHY JR., B.S. in Electrical Engineering

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

May 2014

Acknowledgements

The work this material is based upon has been supported by the National Energy Technology Laboratory of the U.S. Department of Energy under Grant Number DE-FE0002407 and the Computational Science Program at the University of Texas at El Paso.

Abstract

Models of two-phase flow in porous media are of practical interest in many fields of science and engineering such as in oil recovery and hydrology. For this class of problem, an immiscible fluid is injected into a single-phase, porous medium to evict its valuable, native fluid. Because the aim is to amass as much of the indigenous fluid as possible, models attempt to output reliable flow regime patterns depicting how well the invading fluid displaces the native fluid.

Flow regime patterns generally depend upon the geometry of the domain, stimulation at the boundaries, and properties of the invading and native fluids. However, solving for the displacement using the set of differential equations known as the Navier-Stokes equations is not guaranteed at the pore scale. As an attempt to make the problem tractable, the void space of a porous medium where fluid resides is often viewed as an interconnection of simple geometries. The medium then reduces to a graph of nodes and edges, ideally allowing simulators to model flow behavior in larger domains with less computational effort compared to other approaches.

Due to the nature of the simplification, flow in these networks resembles flow in electric circuits. Consequentially, the boundary conditions and simple geometries comprising the system are analogous to electric circuit components. Motivated by the prospect of handling time-varying and/or position dependent boundary conditions, a simulator based upon these electrical equivalents is presented. Because the resulting fluid circuit abides by the conservation laws of electric circuits, a commonly employed method to solve electric circuits in simulation software becomes the basis for an algorithm to arrive at the flow regime patterns.

After providing relevant background information and establishing the translation between a porous medium and its boundary conditions into a circuit equivalent, this work documents all aspects of a two-phase model such that the algorithm is replicable. Such aspects include the network data structure, a set of algorithmic flow rules, a time stepping mechanism, and nodal pressure solver. Attempting to replicate known flow regime patterns for a specific problem where the qualitative behavior is known, this work concludes by outlining the aspects of the model that need refinement according to the results from testing.

Table of Contents

List of Figures

Chapter 1: Summary of Content

Flows in porous media are of interest in many fields of science and engineering with numerous practical applications such as in oil recovery and hydrology. For example, in oil recovery, currently thirty to sixty percent of the original oil-in-place is recovered after enhanced oil recovery techniques (Office of Fossil Energy, 2014). While a significant increase from primary and secondary oil recovery techniques, interest ensues to improve oil recovery methods to extract more oil. Therefore, reliable modeling techniques are instrumental to any hope of improvement.

For this class of problem, an immiscible fluid is injected into a single-phase, porous medium to evict its valuable, native fluid. Because the aim is to amass as much of the indigenous fluid as possible, models attempt to output reliable flow regime patterns depicting how well the invading fluid displaces the native fluid.

To intelligently exploit any system, there needs to be a sense of what aspects of the problem can be manipulated. In the case of enhanced oil recovery, three major techniques to extract more oil are employed to produce more of the reservoir's original oil-in-place with varying degree of commercial success. Thermal recovery, gas injection, and chemical injection each involve manipulating fluid properties to improve flow through the reservoir. However, each technique is relatively expensive and in some cases, unpredictable in its effectiveness. Therefore, other sources for improvement should be explored.

Because there ideally is knowledge about some boundary conditions and control over others, the potential to extract more native fluid by better capturing known conditions and tinkering with controllable ones should be an attractive avenue of study. In this spirit, the objective of this work is to outline a model which handles boundary conditions in a less restrictive manner than the approaches taken by previous network simulators. Using a commonly employed method to solve electric circuits in simulation software, this work handles boundaries using existing analogies between fluid networks and resistive circuits and later discusses which aspects of this approach need refinement according to the results from testing.

Chapter 2 provides background material on porous media, defines the computational domain of interest for the study, and explains how boundary conditions are typically applied in this representative network of the actual domain. Chapter 3 provides the electrical engineering approach to handling a network stimulated by mixed boundaries to arrive at nodal pressures. Although time-varying or position dependent injection or pressure conditions are not considered, this material serves as the framework to handle such conditions. Chapter 4 presents the data structure that tracks the fluids in the network as well as a set of algorithmic flow rules that govern how the fluids mix inside the system. Chapter 5 presents the results from simulations while Chapter 6 concludes by discussing the validity of the model.

Chapter 2: Background

2.1 Introduction

As with other physical phenomena, two-phase flow in porous media is investigated from multiple perspectives to include experimental observation, theoretical formulation, and computational modeling (Joekar-Niasar and Hassanizadeh, 2012). Because solving for the displacement using a set of differential equations is not guaranteed at the length scale of interest, this work focuses on a computational model that reduces a domain into a network of cylindrical tubes and effectively solves a resistive network governed by a set of flow rules. This chapter presents relevant background information on porous media and the computational domain of interest for the study. Later chapters contribute other aspects to this model and give insight into how it is an extension of network analysis from electrical engineering and graph theory, which proves useful in the treatment of boundary conditions.

2.2 Porous Media

A porous medium is a solid material containing voids. The solid part of the material is known as the solid matrix and the remaining void space is occupied by one or more fluids such as oil, water, or gas. Common examples of porous material include soil, sand, fissured rock, cemented sandstone, Karstic limestone, ceramics, foam rubber, bread, lungs and kidneys (Bear and Bachmat, 1991). To be classified as a porous medium, any fluid occupying the continuous void space must be able to flow. An example two-dimensional view within some domain boundary of a porous medium is shown in Figure 2.1. In the figure, the α-subdomain depicts the void space where fluid is allowed to flow while each β-subdomain depicts the solid part of the medium. Specific examples of porous media are shown in Figure 2.2 (Phillips, 1991).

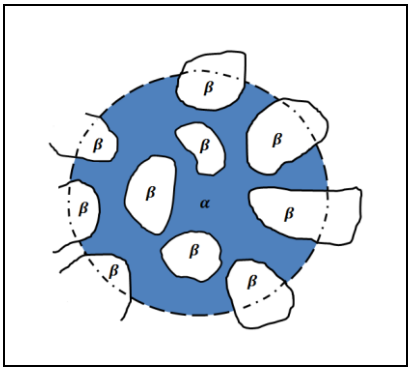


Figure 2.1: Domain.

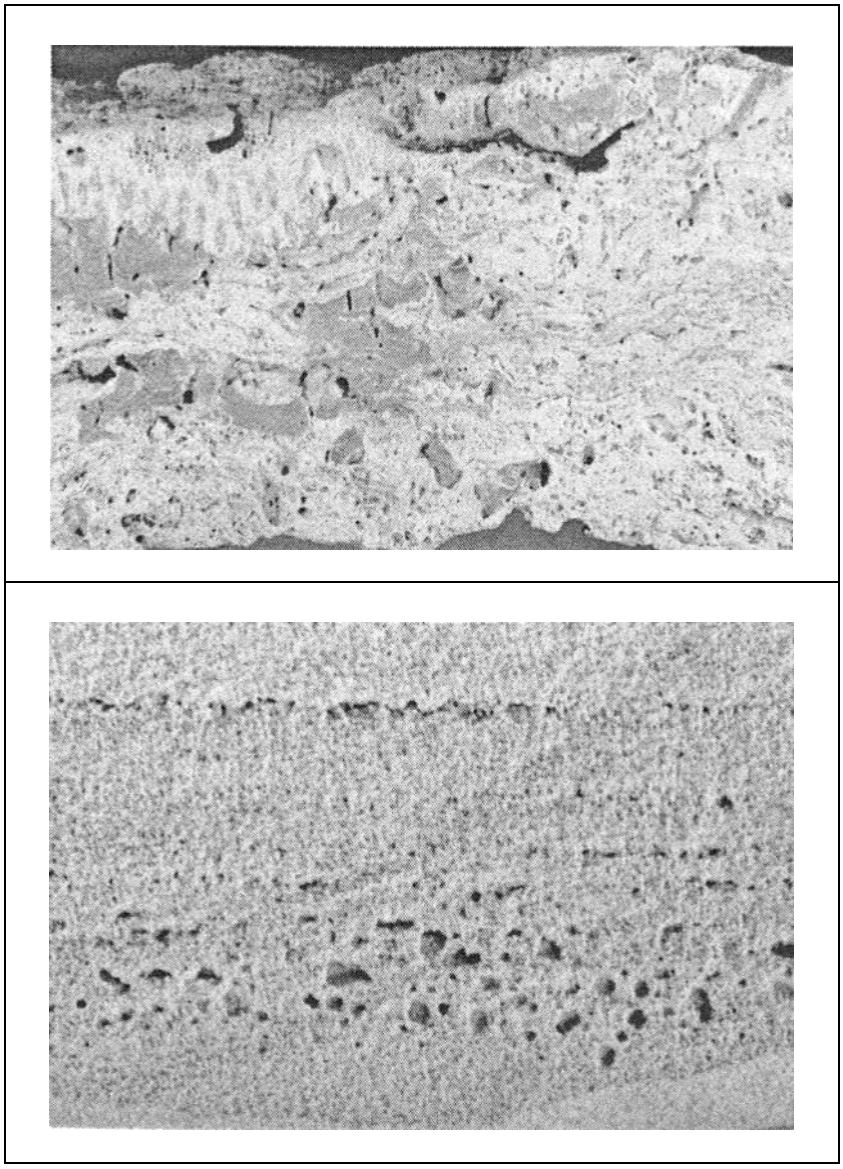


Figure 2.2: (a) Limestone. (b) Partially cemented sandstone.

2.3 Phase

Flows in porous media involve either miscible displacement or immiscible displacement (Petersen, 2009). In miscible displacement, two or more fluids are completely soluble in each other, meaning that there are no distinct interfaces among the fluids contained within the void space. In this case, the flow can be modeled as a single fluid flow and is referred to as a single-phase system. In the immiscible case, an interfacial tension exists among the fluids in the system and distinct interfaces separate the fluids within the porous medium domain. For example, oil and water do not mix and maintain a distinct boundary between each other. Porous media with two or more fluids separated by distinct interfaces are referred to as multiphase systems. This thesis focuses on two-phase systems.

2.4 Length Scales

Predicting the response to some excitation is essential to intelligently exploit any system. To gain the necessary detail to make such management decisions, models are needed to forecast the flows in porous media. Because the flows in these systems can be viewed at different length scales, the amount of detail needed to predict the response of the system depends on the length scale of interest.

A porous structure can be described at four different length scales (Sahimi, 1995). The first scale, which is discernible only through scanning electron microscopy or thin sections, is the pore, or microscopic, scale. Fluid flow at the pore scale is given by the Navier-Stokes equations. Except for only trivial cases, the equations cannot be solved due to complex boundary conditions at the interfaces between the fluids and between each fluid in contact with the solid matrix. The next scale is the core, or macroscopic, scale. A core of rock is taken from a reservoir where empirical correlations are developed from laboratory data using known fluid and rock properties. Immediately following the core scale is the megascopic scale. This scale represents the entire reservoir and is modeled as a collection of thousands or millions of cores. The final scale is the gigascopic scale and is encountered in landscapes that may contain several reservoirs. This thesis focuses its models on the pore scale.

2.5 Displacements

When one immiscible fluid displaces another immiscible fluid in a capillary tube, the fluid for which the contact angle between the tube and the meniscus is smaller than 90° is referred to as the

wetting fluid (Lenormand and Zarcone, 1989). The other fluid in the tube is referred to as the non-wetting fluid. The wettability of the displacing fluid in a two-phase system classifies the type of displacement in the medium. When the wetting fluid displaces the non-wetting fluid in a two-phase system, the process is referred to as imbibition. Alternatively, drainage displacement refers to the process where a non-wetting fluid displaces a wetting fluid in a porous medium. This thesis focuses on drainage displacements.

2.6 Pore Scale Network Models

In pore scale network modeling, the porous medium is an idealized network of simple geometries. While this idealization in general leads to loss of geometrical and topological information, simplification in the medium allows simulators to model flow behavior in larger domains with less computational effort compared to other approaches (Joekar-Niasar and Hassanizadeh, 2012). This section defines a network model as it applies to a porous medium and explains how existing models specifically treat boundary conditions. Figure 2.3 shows a small region of an example two-dimensional porous medium and how to represent it as an idealized network of simple geometries.

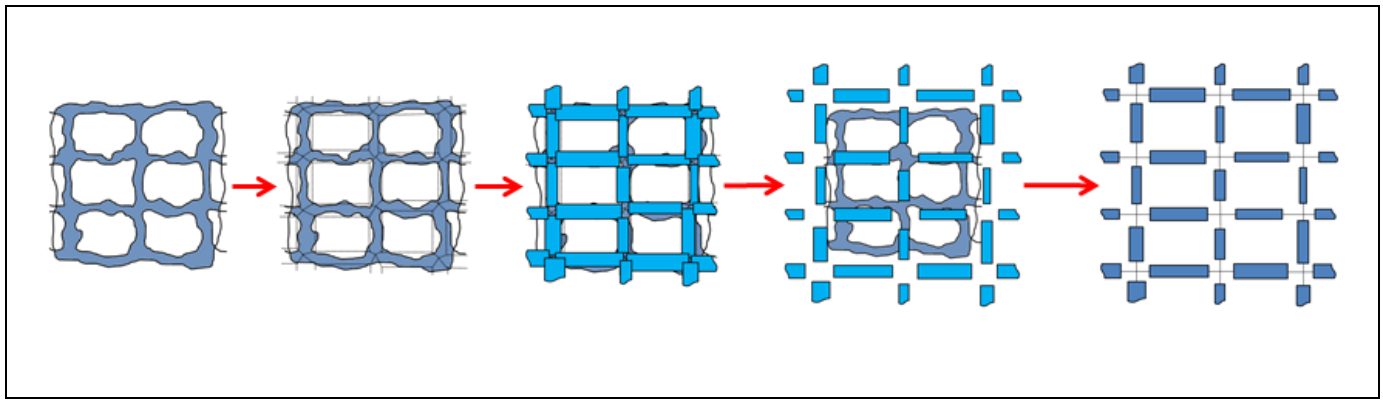


Figure 2.3: Evolution of a region into a network.

2.6.1 The Network Perspective

In general, a flow network is a directed graph of nodes and edges where material flows (Cormen et al., 2009). The material moves through the system from one or more source nodes where it is produced to one or more sink nodes where it is consumed. A flow network assumes an infinite supply of material at each source and that sinks can consume all material it receives. Each edge serves as a conduit for the material to flow with a stated capacity and because each node may be associated with multiple edges, nodes serve as conduit junctions.

In the context of two-phase flow, the nodes and edges in a flow network are usually assigned a geometric volume to approximate the void space of a porous medium. In this fashion, each geometric volume sets the capacity for the material in the network. Unlike most flow network problems, the network is assumed to be filled to capacity at the onset; each node and edge is completely filled with wetting fluid before any source node introduces non-wetting fluid into the system at $t \geq 0$. Similar to an electrical circuit, external excitation and physical properties of the materials occupying the network govern the direction and magnitude of flow in each edge. Because the phases in the network contend to occupy space within a network already filled to capacity, any injected volume of invading fluid forces fluid closest to the sinks to exit the network. However, the interaction of fluid within the network leading up to the sinks is not trivially explained and under this simplified view of a porous medium, a set of algorithmic flow rules are needed to describe how the fluids mix.

Because the flow network perspective is an attractive alternative to the Navier-Stokes equations, there is an inherent assumption that the pressure within an edge is uniform (Ferer et al., 2002). Therefore, these models tend to solve for pressures at the nodes for a network under some pressure and/or flux boundary conditions and use these nodal pressures to advance the fluids in the network according to the aforementioned flow rules.

The flow networks considered in this work treat edges as cylindrical tubes which meet at volume-less nodes. In addition to this geometrical constraint, edges are not allowed to be connected in parallel to easily accommodate the data structures used to represent the network. Regardless, this choice is physically reasonable because, as exemplified in Figure 2.1, more than two edges connected in some

nonparallel configuration are needed to approximate the α-subdomain around a β-subdomain in the domain boundary of the porous medium.

2.6.2 Boundary Conditions

Boundary conditions considered in flow networks are usually constant pressure or constant flux. While implementing a constant pressure boundary condition is numerically straightforward, most pore scale network models use constant flux boundary conditions to study two-phase flow for prescribed capillary numbers. Despite the frequency of this choice, applying constant flow rate conditions at the boundaries has not been applied trivially (Joekar-Niasar and Hassanizadeh, 2012).

Considering the Washburn equation, Aker et al. (1998) equated the total flux over the whole domain as the sum of a function of the global pressure difference and a function of the capillary pressure. Assuming a linear relationship between the pressures at upstream and downstream boundaries, they arrived at a system of equations depending upon the fluid configurations which needed to be solved for two different pressure differences imposed at the boundaries to infer the flux. However, this approach involves solving the pressure field twice at each time step. To avoid this downside, another approach taken by Al-Gharbi and Blunt (2005) assumes that the pressure drop to maintain a constant injection rate changes minimally between successive time steps. According to their perspective, the nodes and centers of the edges admit a pressure. Applying volume conservation at these sites, expressions for the pressures at the nodes and centers of edges are derived and the resulting systems of equations are solved for the nth time step. Designating this global pressure difference as $(\Delta P)_n$, the total injection for the nth time step Q_n is arrived at by summing the flow rates at the inlet edges. Then adopting their assumption, the pressure at the next time step $(\Delta P)_{n+1}$is computed from

$$(\Delta P)_{n+1} = (\Delta P)_n \times \left[1 + \omega \left(\frac{Q - Q_n}{Q}\right)\right], \qquad [2.1]$$

where Q is the desired injection rate and ω is a constant parameter set to 0.5 for their tests.

Regardless of the specifics, each approach is indirect and often computationally expensive. Each approach represents one or more artificial constraints imposed on the network and the specific aim of

this work is to propose and test an alternative which imposes less restraint on the networks that can be studied.

2.7 Hagen-Poiseuille Equation

The Hagen-Poiseuille equation describes the flow of an incompressible, Newtonian fluid in a cylindrical duct (Aker et al., 1998). It is given by

$$q_{i,j} = \frac{\pi r_{i,j}^4}{8\mu d_{i,j}} \Delta P_{i,j}, \qquad [2.2]$$

where the subscripts i and j denote the ends of the tube, $q_{i,j}$ is the volumetric flow rate, $r_{i,j}$ is the radius of the tube, $\Delta P_{i,j}$ is the pressure drop across the tube, μ is the viscosity of the fluid, and $d_{i,j}$ is the length of the tube.

2.8 Capillary Pressure

In the case of immiscible displacement, an interfacial tension exists among the fluids in the system which prevents the fluids from mixing with each other. Consider a tube containing two immiscible fluids as shown in Figure 2.4.

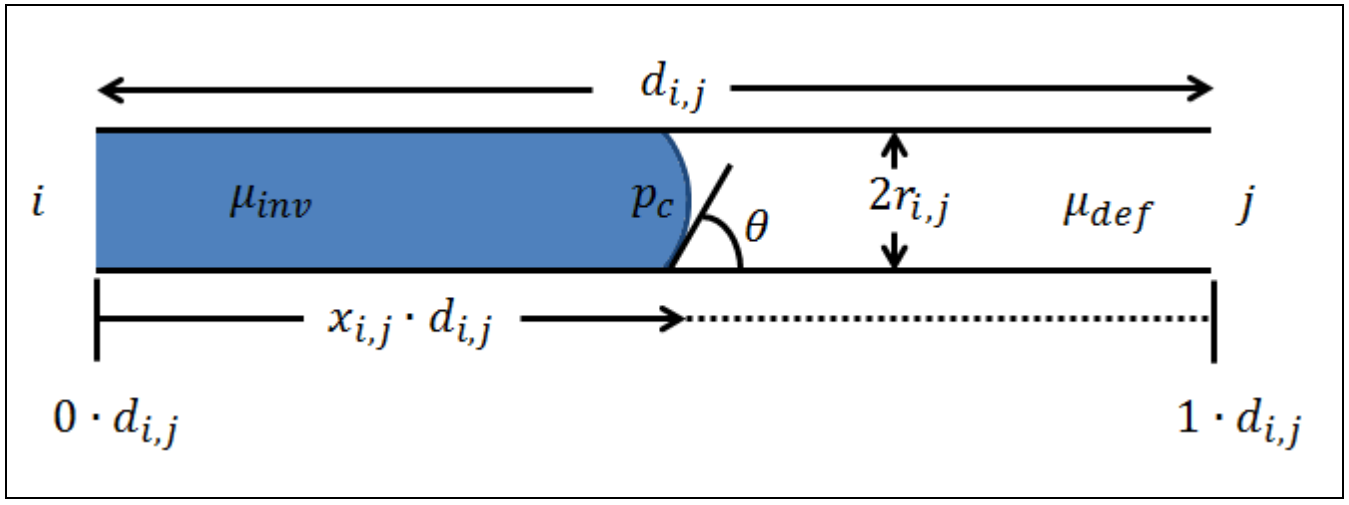


Figure 2.4: Flow in tube containing a meniscus.

The capillary pressure p_c due to the interface between the two phases is given by the Young-Laplace equation

$$p_c = \frac{2\gamma}{r}\cos\theta, \qquad [2.3]$$

where γ is the interfacial tension between the two phases, r is the radius of the tube, and θ is the contact angle between the wetting and non-wetting phases (Aker et al., 1998).

The pressure difference at the ends of the tube must overcome capillary pressure in order for the meniscus to advance further into the tube.

2.9 Washburn Equation

Consider the two-phase tube of Figure 2.4. The Washburn equation describes the flow in this fluid configuration and is effectively an extension of the Hagen-Poiseuille equation. It is given by

$$q_{i,j} = \frac{\pi r_{i,j}^4}{8\mu_{i,j_{eff}} d_{i,j}} \Delta P_{i,j}, \qquad [2.4]$$

where $\Delta P_{i,j} = P_i - P_j - p_{c_{i,j}}$ and $\mu_{i,j_{eff}}$ is the effective viscosity of the tube given by

$$\mu_{i,j_{eff}} = x_{i,j}\mu_{inv} + (1 - x_{i,j})\mu_{def}. \qquad [2.5]$$

In the expression for effective viscosity, μ_{inv} denotes the viscosity of the invading fluid, μ_{def} denotes the viscosity of the defending fluid, and $x_{i,j}$ denotes the fraction of distance into the tube that the invading fluid occupies. Although not explicitly stated in the expression for $q_{i,j}$, the Washburn equation yields the result of the right-hand side of the expression if the arithmetic results in a positive value and zero if the right-hand side produces a negative value (Petersen, 2009).

2.10 Backflow

In drainage displacement, forward flow refers to the non-wetting fluid displacing the wetting fluid in the system. While globally true for the system, in general there may be local instances due to the pressure distribution within the system where a wetting fluid displaces the non-wetting fluid. This occurrence is known as backflow because the interface is forced to retreat rather than continue its invasion (Ferer et al., 2002).

2.11 Flow Regimes

The material in this section is relevant to model validation or rejection. Because this material is specific to a network excited by a single source of constant injection at a lone source node, there is no all-encompassing validation that comes even if some model qualitatively matches these flow regime patterns. However, without specific experimental data for other combinations of circuit excitation for comparison, this is the best available option.

Disregarding the influence of gravity, viscous and capillary forces govern flow behavior during two-phase displacement. Two dimensionless quantities, the capillary number and viscosity ratio, describe displacement with respect to these forces (Petersen, 2009).

The capillary number C_a describes the competition between capillary and viscous forces and is defined as

$$C_a = \frac{Q\mu_2}{\Sigma\gamma\cos\theta}, \qquad [2.6]$$

where Q is the injection rate, μ_2 is viscosity of the non-wetting fluid, Σ is the cross-sectional area of the inlet, γ is the interfacial tension between the two phases, and θ is the wetting angle between the non-wetting and wetting phases. Large viscous dominated flow is observed when this quantity is large while capillary dominated flow occurs when it is small.

The viscosity ratio M is the viscosity of the invading non-wetting fluid divided by the viscosity of the defending wetting fluid, or

$$M = \frac{\mu_2}{\mu_1}. \qquad [2.7]$$

Flow is classified into three regimes: stable, viscous fingering, and capillary fingering. The Lenormand diagram, shown in Figure 2.5, depicts how the capillary number and viscosity ratio relate to each regime (Lenormand, 1990). Outlined by approximate values for $\log M$ and $\log C_a$, each colored region in the figure illustrates how well the invading non-wetting fluid displaces the native wetting fluid in a porous medium. Each of these visuals depicts a single injection site in the center of a two-dimensional porous medium such that the front is allowed to grow outwardly in all directions. During stable displacement, the viscosity of the injected fluid drives the flow as capillary effects and the pressure drop in the displaced fluid are negligible. The flow pattern resembles a mostly flat front with minor instances of trapped wetting fluid behind the front. During viscous fingering, the viscosity of the displaced fluid is the primary mechanism driving the flow as capillary effects and the pressure drop in the displacing fluid are negligible. The flow pattern resembles tree-like fingers with no loops that spread across the whole network and grow towards the outlet. During capillary fingering, the viscous forces are negligible in both fluids and capillarity drives the flow. The fingers in this pattern grow in all directions and form loops which trap the displaced fluid. As shown in the figure, stable displacement maximizes displacement effectiveness and either fingering regime is unfavorable.

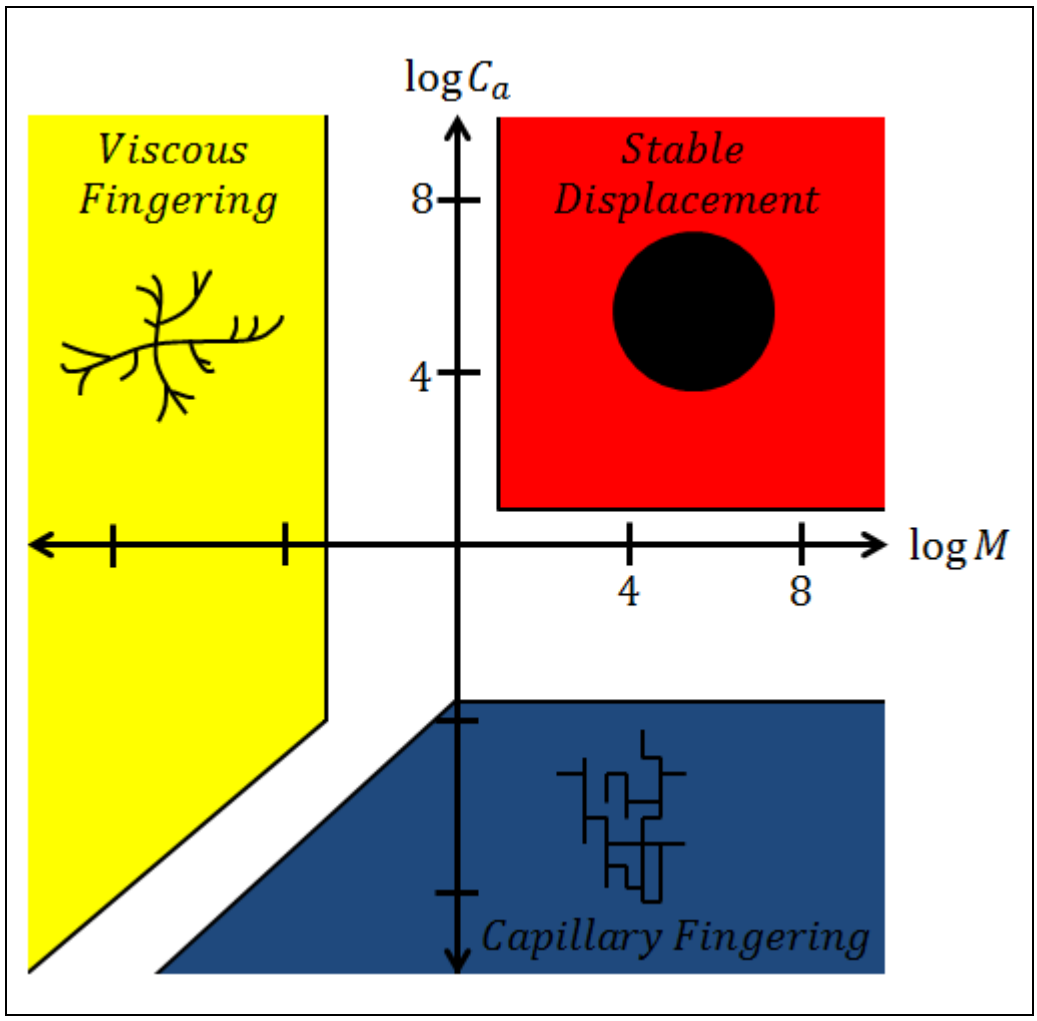


Figure 2.5: Lenormand diagram.

Possibly trivial, the cross-sectional area of the inlet Σ is usually estimated as the product of the width of the network by a thickness equal to the distance between two nodes. However, because the cross-sectional area of each inlet tube is known, this work deviates from this practice and computes the value as the sum of the cross-sectional area of each inlet tube. Although no attempt is made at locating the boundaries of each flow regime region, this choice would amount to renumbering the scale of the $\log C_a$ axis in Figure 2.5 as Σ would not be an overestimation of its true value.

Chapter 3: Analysis of Circuits

3.1 Introduction

A simplified pore scale network model reduces a porous medium into an interconnection of tubes that meet at nodes. Because solving for the displacement using the Navier-Stokes equations is not guaranteed at the pore scale, this brand of model circumvents solving the set of differential equations altogether by solving for the pressures at the nodes and updating the displacement according to the resulting pressure distribution and a set of algorithmic flow rules. In this fashion, the network is analogous to an electric circuit.

To handle boundary conditions more effectively than known models, this chapter provides the necessary information to understand how to use computer-aided analysis techniques to solve the pressures in pore networks. These techniques are based on basic electrical circuit theory and used quite extensively in software such as Simulation Program with Integrated Circuit Emphasis, or SPICE. As such, the information in this chapter is presented using electric circuits. The chapter concludes by describing the translation between a porous medium and its electrical equivalent such that the application of these techniques is straightforward.

3.2 Circuit Overview

An electric current consists of charges in motion from one region to another (Young et al., 2004). When an electric current occurs within a conducting path that forms a closed loop, the path is called an electric circuit. Essentially, electric circuits convey energy from one place to another. As charged particles move within a circuit, electric potential energy is transferred from a source to a device in which that energy is either stored or converted to another form. If the energy is consumed in the device, it is known as a resistor.

The concept of electric charge is the basis for describing all electrical phenomena. In circuit theory, the separation of charge creates an electric force, or voltage, while the motion of charge creates an electric fluid, or current. Although current involves discrete moving electrons, the electrons are not considered individually. Because an enormous number of electrons are present, electrons and their

corresponding charge are considered as one smoothly flowing entity. Therefore, current is considered as a continuous variable.

Circuits are constructed with components (Nilsson and Riedel, 2008). Examples of components include batteries, resistors, inductors, capacitors, diodes, transistors, transformers, and operational amplifiers. Each component satisfies a mathematical model of the actual electrical component. Due to the nature of circuit models, components are modeled in terms of the voltage and current at its terminals. Components are connected with conducting material to form one or more closed loops and are drawn diagrammatically using a schematic. The combination of components and interconnections allows for simple and complex operations to be performed thereby making the circuit useful.

Circuit analysis is based on mathematical techniques used to predict the behavior of these components and interconnections. Pencil and paper, calculator, and computer methods are generally available for performing the calculations of circuit analysis. Because the enormity of networks make hand calculations intractable, reliable computer methods are desirable.

Computer-aided analysis of electronic circuits involves two stages (Ioinovici, 1990). The first stage involves defining a finite group of components. The second stage of computer-aided analysis involves describing the interconnection of the defined circuit components.

3.3 Relevant Circuit Elements

For the purpose of relating analysis of electric circuits to porous media, it is commonly assumed the voltage source, current source, and resistor can be related to a fluid network. Voltage and current are analogous to pressure and injection rate, respectively. Because the tubes admit a conductance, the resistor is taken to be analogous to the fluid-filled tubes in the two-phase fluid network.

An electrical source converts nonelectric energy to electric energy and vice versa (Nilsson and Riedel, 2008). For example, a battery in use converts chemical energy to electric energy whereas a battery being charged converts electric energy to chemical energy. The sources considered within this report are ideal; there are no internal resistive, capacitive, or inductive parasitic effects. Additionally, the sources considered are independent. An independent source is one that always produces its labeled voltage or current regardless of the load presented to it whereas a dependent source establishes a voltage

or current whose value depends on the value of voltage or current elsewhere in the circuit. The circuit symbols for an ideal, independent voltage and ideal, independent current source are shown in Figure 3.1.

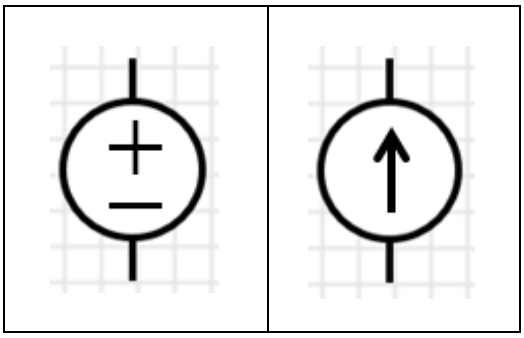


Figure 3.1: Circuit symbols. (a) Voltage source. (b) Current source.

The resistor is a circuit element which impedes the flow of electric charge. The amount of opposition to flow depends on the material. A common circuit symbol for the resistor is shown in Figure 3.2.

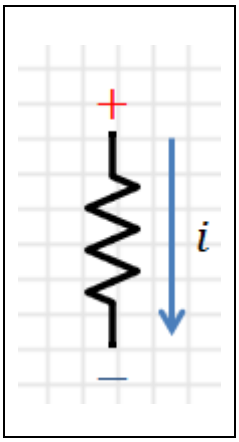


Figure 3.2: Circuit symbol for resistor.

As shown in Figure 3.2, this work adopts the convention that current flows in the direction of the voltage drop across the resistor. Ohm's law, which describes the relationship between the voltage difference v across its terminals and current i through the element in terms of its resistance R, is given by

$$v = Ri. \qquad [3.1]$$

Alternatively, in terms of conductance, or the reciprocal of resistance, G, Ohm's law is sometimes given by

$$i = Gv. \qquad [3.2]$$

3.4 Interconnections

Interconnections are described in terms of nodes and loops. As discussed before, a node in a circuit is a point where two or more circuit elements meet. A loop is any closed conducting path.

Although general interconnections exist, two common configurations are known by name. Envision a collection of resistors and let each one have one terminal named a and another terminal named b. If these resistors are connected in such a way that all a terminals connect to a single node and all b terminals connect to a single node, then the resistors are said to be connected in parallel. When resistors are connected in sequence, the resistors are said to be connected in series. Examples of these configurations are shown in Figure 3.3.

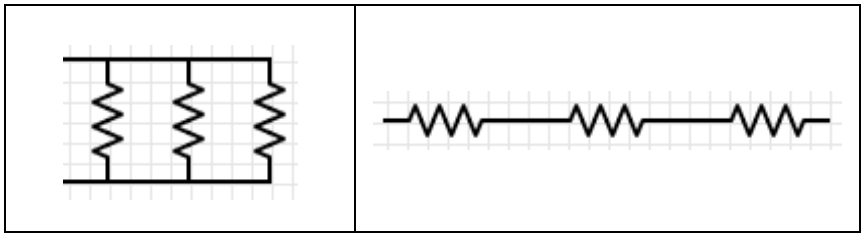


Figure 3.3: Resistor configurations. (a) Parallel. (b) Series.

3.5 Kirchhoff's Rules

A circuit is said to be solved when the voltage across and the current through every element has been determined. While Ohm's law often contributes to deriving such solutions, it is in general not

enough even for trivial circuits with only sources and resistors interconnected in a network. General methods exist for analyzing circuits with resistors connected in series or in parallel, but for networks with more complex resistor configurations, two rules known as Kirchhoff's rules are needed. Kirchhoff's rules are really two equilibrium equations (Nilsson and Riedel, 2008). One is based on the principle of conservation of charge applied to a node. The other is derived from energy conservation of charge moving around a closed loop.

Because no charge can accumulate at a node, the total charge entering the node per unit time equals the total charge leaving per unit time. Charge per unit time is electric current. Therefore, if current entering a node is taken to be negative and current leaving a node to be positive, the sum of currents into a node must equal zero. Kirchhoff's node rule is often referred to as Kirchhoff's current law, or KCL, and is given by

$$\sum i = 0. \qquad [3.3]$$

The loop rule is a statement that voltage is conservative. Therefore, the algebraic sum of voltages around a loop, adopting some sign convention, must equal zero. If this were not true, the voltage across an element in the loop does not have a definite value. The loop rule is often referred to as Kirchhoff's voltage law, or KVL, and is given by

$$\sum v = 0. \qquad [3.4]$$

3.6 Equivalent Resistance

Suppose an interconnection of resistors exists between nodes a and b. The concept of equivalent resistance essentially determines the resistance seen between a and b. In this fashion, the interconnection of resistors can be replaced with a single resistor of equivalent resistance and ideally not cause issue elsewhere in the circuit.

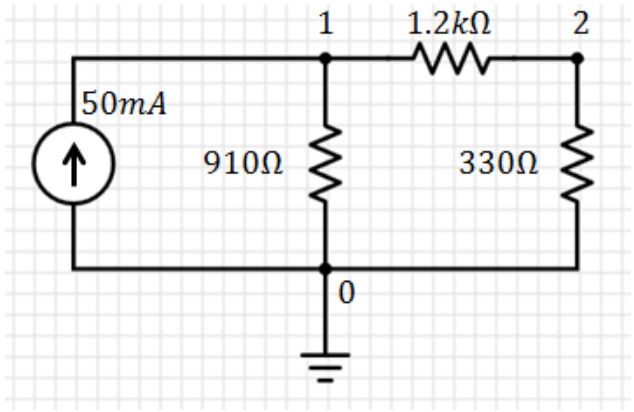
Equivalent resistance usually involves reducing a set of resistors connected in series or parallel combinations into a single resistor. Although not apparent now, for the purposes of this work, the equivalent resistance for resistors connected in series is relevant. Let n resistors, where $n \geq 2$, be connected in series between nodes a and b. Then, the equivalent, or total, resistance between a and b is given by

$$R_{eq} = \sum_{i=1}^{n} R_i = R_1 + R_2 + \cdots + R_n. \qquad [3.5]$$

Deriving this expression is straightforward by connecting a test voltage source between a and b and using KVL on the only loop of the resulting circuit.

3.7 The Netlist

Circuits must be conveyed to the computer. SPICE forms a file known as a netlist to store circuits given to it (Mitcheson, n.d.). Prior to forming the file, each node in the circuit is assigned a number. Because circuits require one or more closed loops, the terminals of each component meet at a numbered node. Although the nodes could be numbered in different ways, the interconnection is unique. A netlist file is effectively a table of the components in the circuit. Each row of this table gives the information regarding the component including its type, value in standard units, and the node numbers each of its terminals connects. Figure 3.4 gives an example network and a netlist to describe the circuit.



I	0	1	0.05
R	0	1	910
R	1	2	1200
R	0	2	330

Figure 3.4: (a) Example network. (b) Netlist.

Notice that ground is numbered zero in the netlist in Figure 3.4. For simplicity, this node is always taken to be node zero in electric circuits. This convention is carried over to the pore networks modeled.

3.8 Nodal Method

Recall the circuits within this study are composed with only independent sources and resistors. Therefore, these circuits are solved when the voltage across and the current through every resistor has been determined. Because voltage at some point is measured relative to another point, the magnitude of the voltage across a resistor is measured across this component's terminals. Therefore, if the voltages at all terminals relative to a common reference node are known, then the voltage across and current through every resistor is available through systematic computation. The common reference node in an electric circuit is ground and schematically is given by one of the symbols in Figure 3.5. Without loss of generality, by convention, ground has a value of zero volts. The nodal method gives the voltages at the non-reference nodes relative to ground and provides insight into understanding how computer programs such as SPICE operate (Ioinovici, 1990). This approach effectively formulates circuit equations using Kirchhoff's current law at each non-reference node. These equations are arranged into the matrix equation

$$GV = I. \qquad [3.6]$$

The conductance matrix G is a square matrix. The number of rows (conversely, number of columns) is equal to the number of non-reference nodes. The vector I represents the sum of currents actively injected into each node. Most of these entries are zero because KCL requires currents into a node to sum to zero. However, an entry is nonzero if one or more current sources are injecting current into a node.

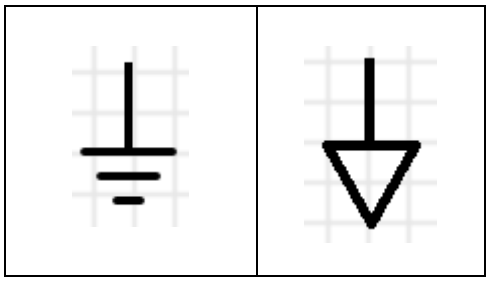


Figure 3.5: Common ground node symbols.

As an example, consider the circuit in Figure 3.4. Before forming the matrix equation, current directions must be specified. It is already known that the current in the current source flows from node 0 to node 1. However, the current through the resistors must be assumed. For this example, assume the current through the 910Ω resistor flows from node 1 to node 0, the current through the $1.2k\Omega$ resistor flows from node 1 to node 2, and the current through the 330Ω resistor flows from node 2 to node 0. At node 1, KCL yields

$$0.05 = \frac{v_1 - 0}{910} + \frac{v_1 - v_2}{1200} \qquad [3.7]$$

while KCL at node 2 yields

$$\frac{v_1 - v_2}{1200} = \frac{v_2 - 0}{330}. \qquad [3.8]$$

Rearranging the individual equations, the matrix equation is found to be

$$\begin{bmatrix} \frac{1}{1200}+\frac{1}{910} & -\frac{1}{1200} \\ -\frac{1}{1200} & \frac{1}{1200}+\frac{1}{330} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0 \end{bmatrix}. \quad [3.9]$$

After forming the matrix equation, the nodal voltages are found by solving for left-hand vector. After these node voltages are known, the voltage across and current through each resistor is easily determined from Ohm's law.

The specific rules for the nodal method are as follows: select a reference node, number the remaining nodes and assign a current direction through each resistor and each current source, apply KCL to each node not connected to a voltage source, and solve the system for the unknown voltages (Cheever, n.d.). However, it is inconvenient from a programming perspective to perform KCL analysis for each arbitrary network with varying components and connectivity to establish the matrix form of the nodal method. Fortunately, patterns emerge based on the elements involved and the nodes each connects. However, because the nodal approach uses KCL exclusively, each source in the circuit must be a current source in order for the circuit to be solved directly using this matrix equation. Due to this limitation, the details to form the matrix equation using these convenient patterns are withheld because the method in the next section, which handles both current and voltage sources, generalizes to the nodal method.

3.9 Modified Nodal Method

Modified nodal analysis, or MNA, directly handles independent voltage and independent current sources. Although MNA often produces a larger system of equations than other methods, it is straightforward to implement in software because patterns, based solely on the elements in the network and the nodes each connects, emerge allowing its matrix equation to be populated in a straightforward manner (Ho et al., 1975). In essence, this method forms a system of equations by writing equations for each node not attached to a voltage source as in the standard nodal method and additional equations for each voltage source. The additional equations are simple to form because any such equation simply states that voltage source equals the difference in node voltages at its terminals. As with the classic nodal method, MNA solves for the voltages at the nodes relative to the reference node. However, MNA

also solves for the currents through the voltage sources. Because these currents are not needed for the fluid networks, these entries are simply ignored.

Let a circuit have m voltage sources and $n + 1$ nodes. To perform modified nodal analysis, these general rules must be followed: select a reference node and number the remaining n nodes, label currents through each current source, assign a name to the current through each voltage source using the convention that current flows from the positive node to the negative node of the voltage source, apply KCL to each node taking currents out of the node to be positive, write additional equations for the voltage across each voltage source, and solve the system for the node voltages and currents through the voltage sources (Cheever, n.d).

MNA results in a matrix equation of the form

$$Ax = z. \qquad [3.10]$$

In general, A is a $m + n$ by $m + n$ matrix with entries to be discussed momentarily, x is a $m + n$ column vector of unknown node voltages and unknown voltage source currents, and z is a column vector of known quantities that is filled out using the current and voltage sources. Fortunately, for the purposes of computer programming, this matrix equation is formed by inspection using so-called "element rubber stamps" to fill in the entries in the square matrix and right-hand vector. To best describe these stamps to fill out A and z, first let the elements in A and z be initialized to all zeroes. Then, when a value is stamped into the matrix equation, the value is summed into the proper location or locations. Let A be partitioned into four submatrices, x into two submatrices, and z into two submatrices. A, where the superscripts in the expression denote the dimensions of each matrix, is

$$A^{(n+m)\times(n+m)} = \begin{bmatrix} G^{n\times n} & B^{n\times m} \\ C^{m\times n} & D^{m\times m} \end{bmatrix}. \qquad [3.11]$$

The entries in the G matrix are determined by the interconnections between resistors. Each element in this matrix is a conductance, or reciprocal of resistance. Each element along the diagonal is

equal to the sum of the conductances of each element connected to the corresponding node; the first diagonal element is the sum of conductances connected to node 1, the second diagonal element is the sum of conductances connected to node 2, etc. The elements not along the diagonal are the negative conductance of the element connected to the pair of corresponding nodes. For example, the negative conductance of a resistor connected between nodes 1 and 2 is stamped into G at the locations $g_{1,2}$and $g_{2,1}$where $g_{i,j}$ represents the element at row i and column j for $1 \leq i,j \leq n$ in the matrix G. Note that there is no row zero or column zero. Therefore, any resistor connected to the reference node contributes only one entry to the G matrix, which is the appropriate location on the diagonal. After considering all resistors, any entries not assigned a conductance remain zero.

The entries in the B matrix are stamped with either a -1 or 1. The entry $b_{i,j}$ is determined by the voltage sources and nodes in the network. If the positive terminal of the ith voltage source is connected to node j, then $b_{i,j}$ is a 1. If the negative terminal of the ith voltage source is connected to node j, then $b_{i,j}$ is a -1. In this fashion, any ungrounded voltage source contributes two entries to the B matrix, a 1 and -1 in the same column. Because the matrix indices begin at one, any grounded voltage source only contributes a single entry to this matrix.

The final two matrices are even simpler to form. Because dependent sources have been excluded from consideration, the C matrix is simply the transpose of the B matrix and the D matrix is composed entirely of zeros.

The vector x, where the superscripts in the expression denote the dimensions of each matrix, is

$$x^{(n+m)\times 1} = \begin{bmatrix} v^{n\times 1} \\ j^{m\times 1} \end{bmatrix}. \qquad [3.12]$$

Each element v_k in the vector v corresponds to the unknown voltage at node k in the circuit. As before, there is no entry for the reference node because the reference node is zero and the row number satisfies the inequality $1 \leq k \leq n$. Each element j_k in the vector j corresponds to the unknown current through each voltage source.

The vector z, where the superscripts in the expression denote the dimensions of each matrix, is

$$z^{(n+m)\times 1} = \begin{bmatrix} i^{n\times 1} \\ e^{m\times 1} \end{bmatrix}. \qquad [3.13]$$

Each element in the vector i corresponds to a particular node. The row number corresponds to the node number and the values occupying the vector correspond to currents. The value at any given row is the sum of current sources into the corresponding node. Obviously, any node not connected to a current source sees no injected current and the appropriate entry in this vector is a zero as a result. For the purposes of forming the proper sum, recall that current exiting a node is positive while current entering a node is negative. Finally, each element in the vector e is equal in value to the corresponding voltage source.

As an example of MNA, consider the network in Figure 3.6. Because the circuit contains a single voltage source and two non-reference nodes, $m = 1$ and $n = 2$. The matrix equation is formed by constructing each submatrix first. The matrix G is formed as in the standard nodal method and is given by

$$G = \begin{bmatrix} \frac{1}{220} + \frac{1}{330} & -\frac{1}{330} \\ -\frac{1}{330} & \frac{1}{330} + \frac{1}{1000} \end{bmatrix} \qquad [3.14]$$

while the matrices B and C^T are determined to be

$$B = C^T = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \qquad [3.15]$$

Because D is trivially the zero matrix, A is completely known. If the current through the voltage source is denoted by i_{V_s}, then the left-hand vector x is given by

$$x = \begin{bmatrix} v_1 \\ v_2 \\ i_{V_s} \end{bmatrix}. \qquad [3.16]$$

Finally, the right-hand vector z, formed from its subvectors, is given by

$$z = \begin{bmatrix} 0.05 \\ 0 \\ 3.3 \end{bmatrix}. \qquad [3.17]$$

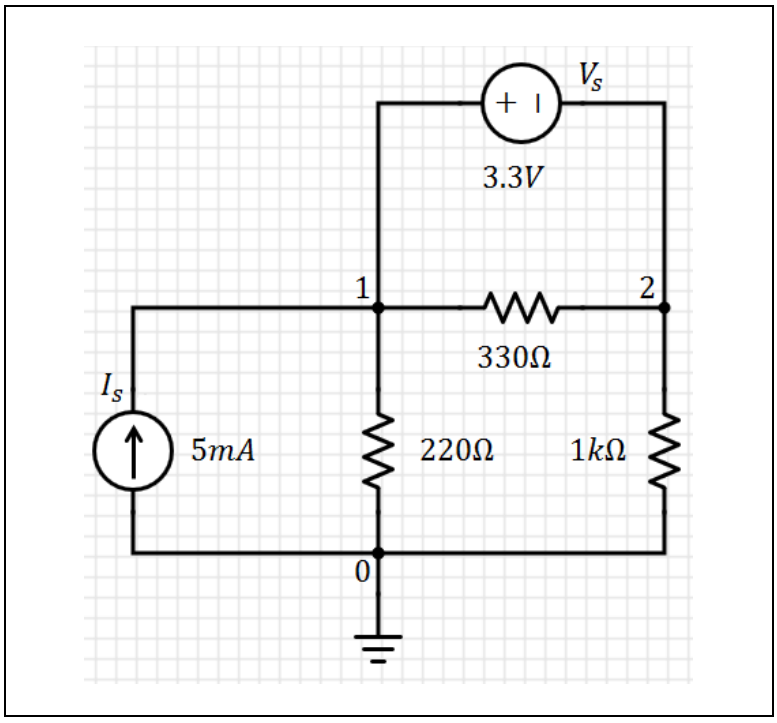


Figure 3.6: Example network.

3.10 Extension of Computer-Aided Analysis to Fluid Networks

To complete the analogy between electric circuit and fluid network, this section outlines how to translate a description of a network and its boundary conditions into a circuit. The resulting schematic really completes the analogy between electric circuits and pore scale networks. Once a schematic is known, modified nodal analysis is used for each time step to solve for nodal pressures in the algorithm of the next chapter that advances the fluids.

Consider the unrealistically small single-phase network in Figure 3.7. As shown, the network is a lattice of tubes perpendicular to one another. To avoid confusion, keep in mind that the angles of the

tubes have no effect on the flow in the circuit; no mathematical expression presented in this work considers the angle of a tube between its two connecting nodes.

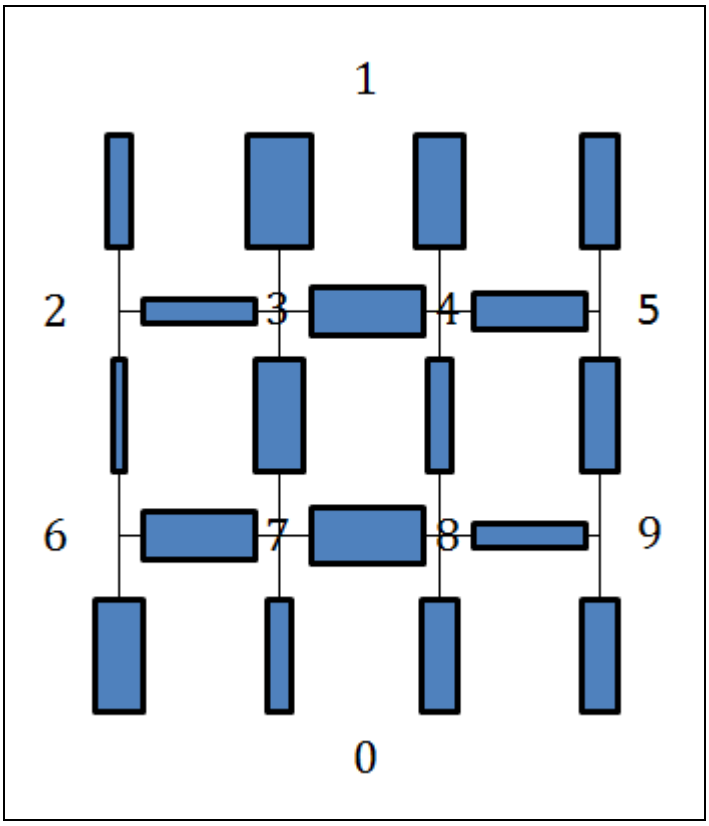


Figure 3.7: Lattice of tubes.

Although not shown diagrammatically, the top row of nodes in Figure 3.7 are all injected with the invading fluid at a constant injection rate. In this fashion, these nodes are all really the same node because a constant injection rate is achieved with a flux source. The bottom row of nodes serve as the outlet and as with the top row, these nodes are all really the same node. The other nodes are connected as shown in the figure. The equivalent schematic diagram is shown in Figure 3.8.

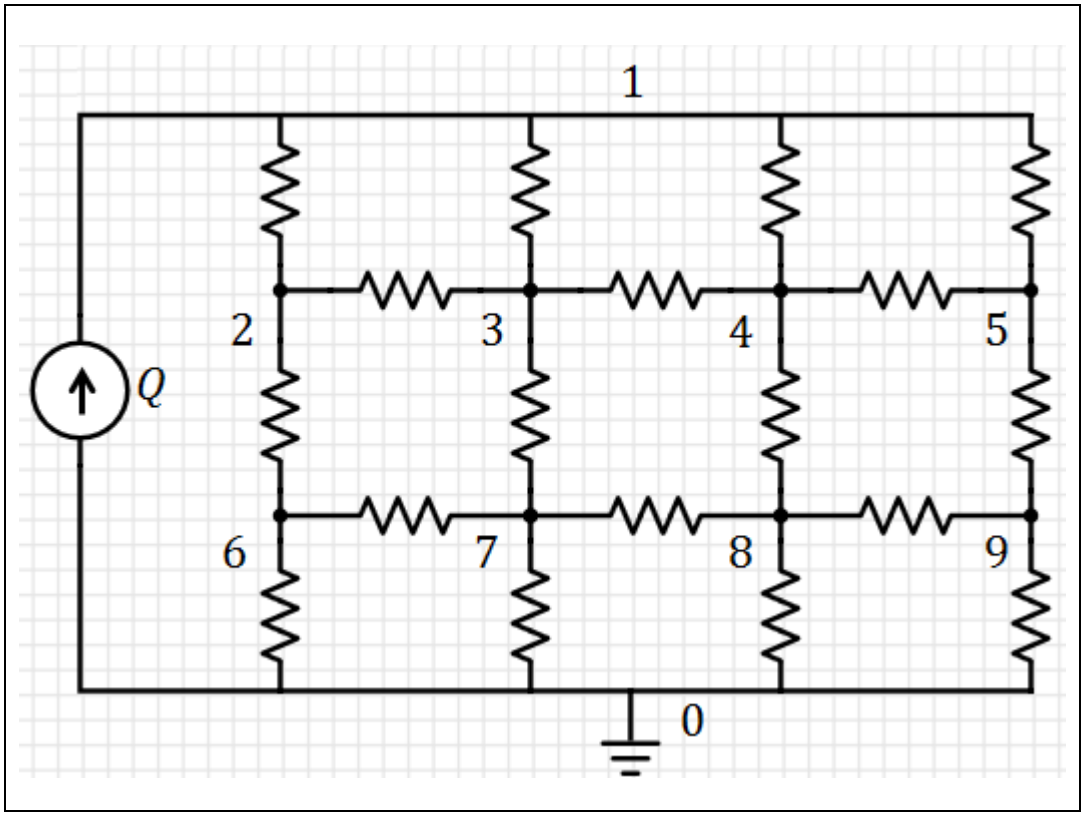


Figure 3.8: Schematic for top row of nodes with constant injection rate.

As shown in Figure 3.8, the outlet node is the reference node. Because the injection rate of the top row matches the eviction rate at the bottom row, the shown schematic is justifiable. Also notice in the figure that the nodes are numbered. Because these fluid circuits must be conveyed to the computer, the nodes are numbered to later aid in the formation of a netlist.

Consider Figure 3.8 again. To model a second situation, now assume that the top node remains a source node where the invading fluid enters. This time, however, let nodes 1, 2, 5, 6, and 9 have pressure conditions relative to the outlet node. The equivalent circuit is shown in Figure 3.9.

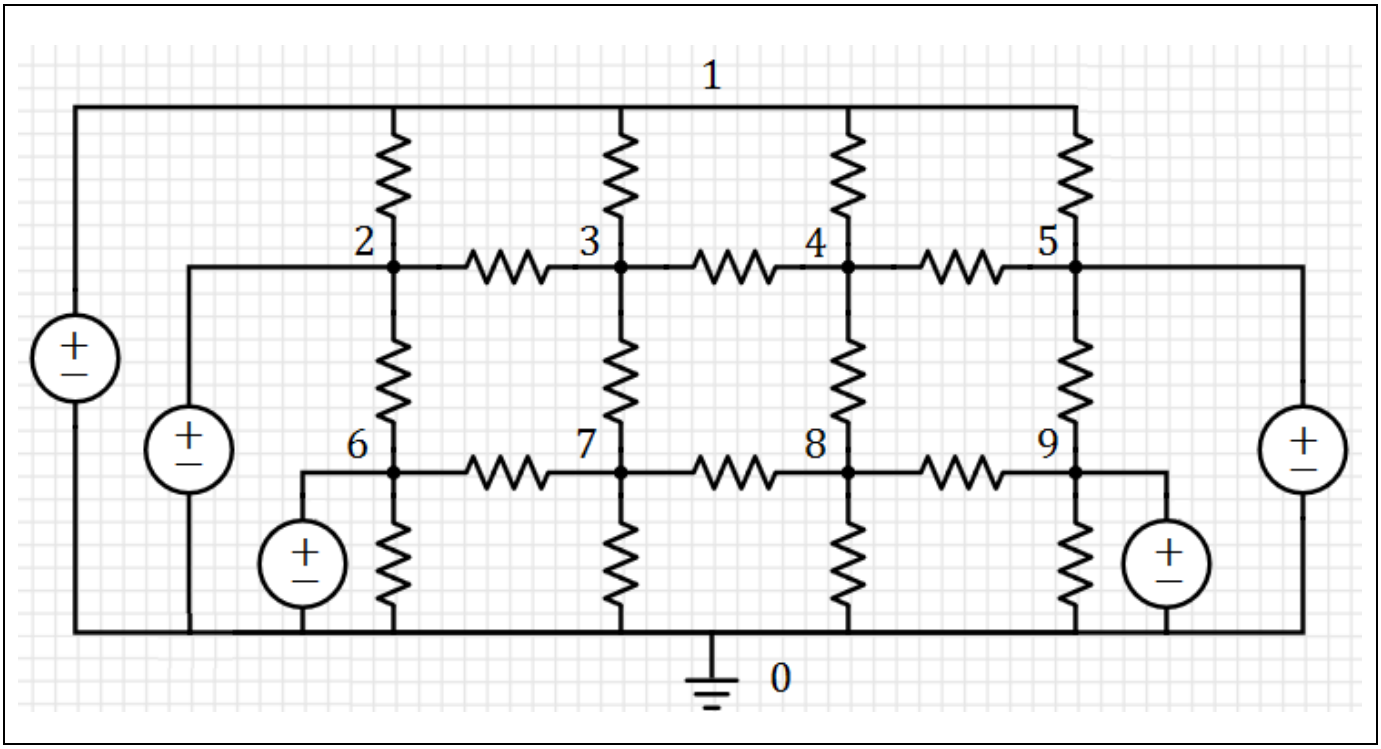


Figure 3.9: Schematic for top row of nodes with constant pressure relative to outlet.

Conceivably, several complicated interconnections of tubes and boundary conditions can be considered if they do not violate the circuit. For example, the schematic in Figure 3.10 shows an invalid combination of boundary conditions. Note that the interconnection of tubes is concealed in a so-called "black box".

Because the translation between a geometric description and schematic is straightforward, further examples of this process are discontinued.

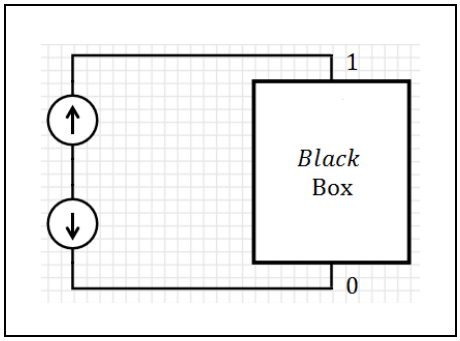


Figure 3.10: Invalid boundaries.

Chapter 4: Algorithm of Flow for Two-Phase Systems

4.1 Introduction

Consider the simplified view of a two-phase system excited by some boundary conditions. Initially, the network of cylindrical tubes, which meet at volume-less nodes, is completely filled with a wetting fluid. Through some external excitation at the boundaries, a non-wetting fluid is injected into the network to evict the native wetting fluid through the outlet boundary. Regardless of the specific stimulation, these sources of excitation along with the resistances of the tubes cause differences in the pressures between any two nodes in the network. Fluid flows between any two adjacent nodes provided the difference in pressure between both nodes exceeds the capillary pressure for those nodes' connecting tube. Under these conditions, the invading fluid displaces the native fluid and the resistance of the network continually changes. While a method has been established at this point to solve for the pressures at the nodes, the matter of documenting how the fluids interact with each other and how to cope with the continually changing resistance of the network remains. Additionally, because tracking every meniscus in even a single tube can prove hopeless, any simplified model is not capable of capturing the exact nature of the flow. Therefore, a set of well-defined rules, which the fluids follow to flow throughout the system unambiguously, must be defined to record as much of the flow as algorithmically possible. Each issue is addressed within this chapter.

4.2 The Network Data Structure

While the data structures tracking the fluids within the network limit what aspects of the flow are recordable, the data structures should still be conducive to networks with irregular connectivity. Consistent with network flow problems, adjacency lists or adjacency matrices are typically used to represent the structure as each meets this criterion (Cormen et al., 2009). Because the networks considered within this report are sparse, or the number of edges is significantly less than the square of the number of nodes, the algorithm is presented using an adjacency list data structure.

In essence, an adjacency list representation is a list of lists. The main list records every node in the network. Each of these entries is associated with a list of its own whose entries are the nodes adjacent to the node in the main list. As can be concluded, an adjacency list is straightforward to form

from a netlist file representing a network of interest. Figure 4.1 shows an example network and its corresponding adjacency list representation. While other implementations for this data structure exist, the figure shows the example adjacency list as an array of pointers to linked list data structures.

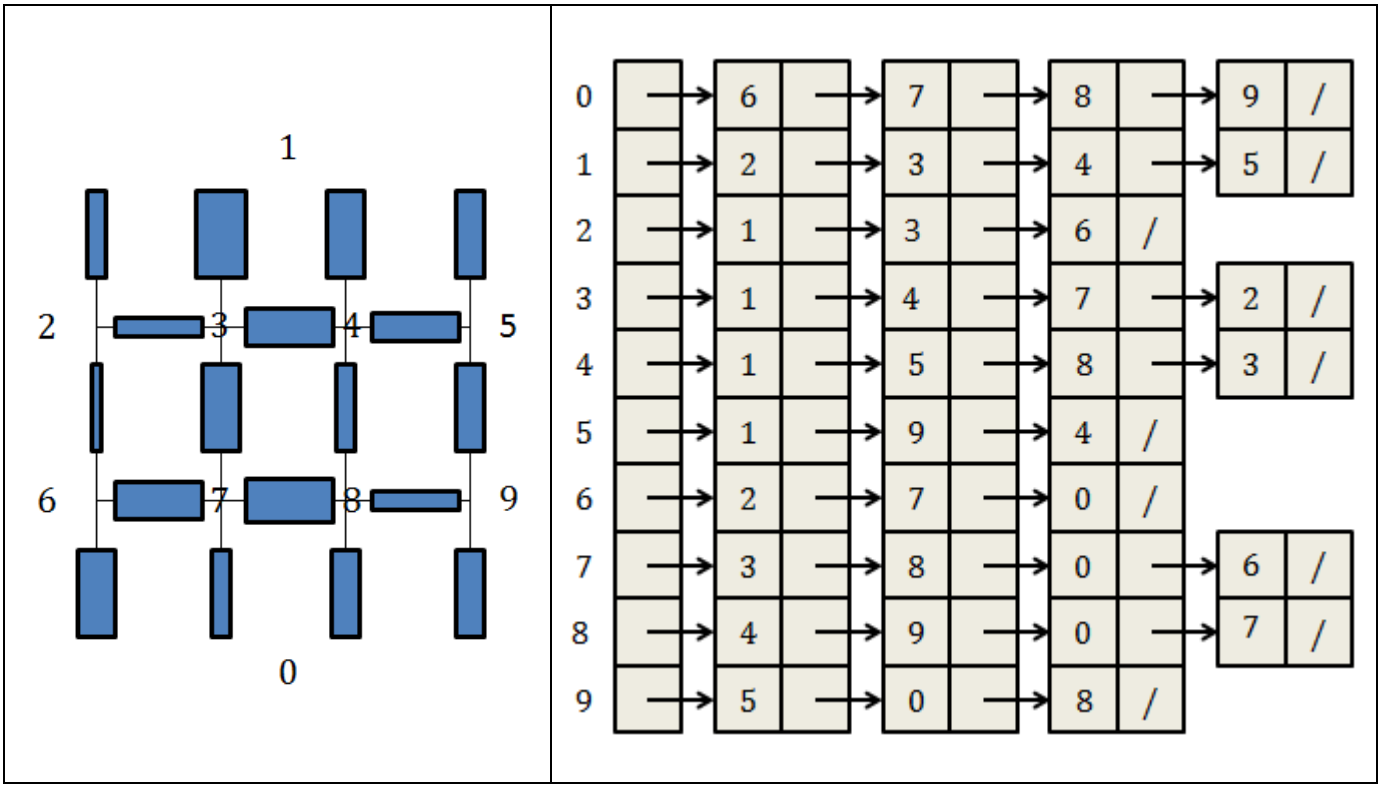


Figure 4.1: (a) Example network. (b) Its Adjacency list representation.

Adopting an object-oriented programming view of the problem, additional information about the nodes and edges is stored in the adjacency list. Using this paradigm, nodes are assigned pressure and type attributes. The pressures stored in this data structure are the result of MNA and must be updated for each progression in time. For every instance of the node class, the type attribute signifies whether or not non-wetting fluid has reached that node object. As can be concluded, the source node is non-wetting while the sink node, because it must allow both fluids to exit the network, is the only node whose type attribute is inconsequential. At $t = 0$, the remaining nodes in the system are wetting and change type as specified by the flow rules to be discussed later in this chapter. Similarly, an edge, which corresponds to a link in the linked lists, is assigned attributes of its own such as length and radius, conductance, and

flux. In addition to these obvious attributes, the structure tracks two other attributes to make sense of which fluids reside inside the tube. These attributes are explained in the next section.

As written, the example adjacency list in Figure 4.1 is undirected. If the main list conveyed in the array of pointers is named Adj, for each node u corresponding to an index of the array, $Adj[u]$ contains all the nodes v such that there is an edge between the node u and each node v. More memory is required to represent the network in the undirected case than the directed case. Because attention must be taken to maintain a consistent view of the attributes in the network, additional looping costs are associated with this implementation. In contrast, $Adj\ [u]$ in a directed graph contains all the nodes v when current flows from u to v. An example directed adjacency list is shown in Figure 4.2 for the accompanying network with specified flow directions. Not to be confused with a source associated with a boundary condition, a node u in the main list depicting this situation is referred to as a source node while each v is referred to as a destination node. Because current flows out of a source node to each of its destination nodes, the edges associated with these nodes are referred to as outgoing tubes. Because the pressure distribution is ever-changing, the directed implementation needs to accommodate changes in current direction. Therefore, there must be the ability to remove an element l from the linked list of $Adj\ [u]$ and insert it into the linked list of $Adj[l]$ during any iteration if it is warranted. Because there is an unlikely need to relocate several elements during an iteration (with the possible exception of the first iteration when the current directions are initially assumed to form the netlist), the directed case is regarded as the best choice. However, because it is straightforward to loop through either form of the data structure to form the G matrix in Equation 3.11 for MNA and update attributes of the network when needed, either form is technically amenable to the algorithm discussed momentarily.

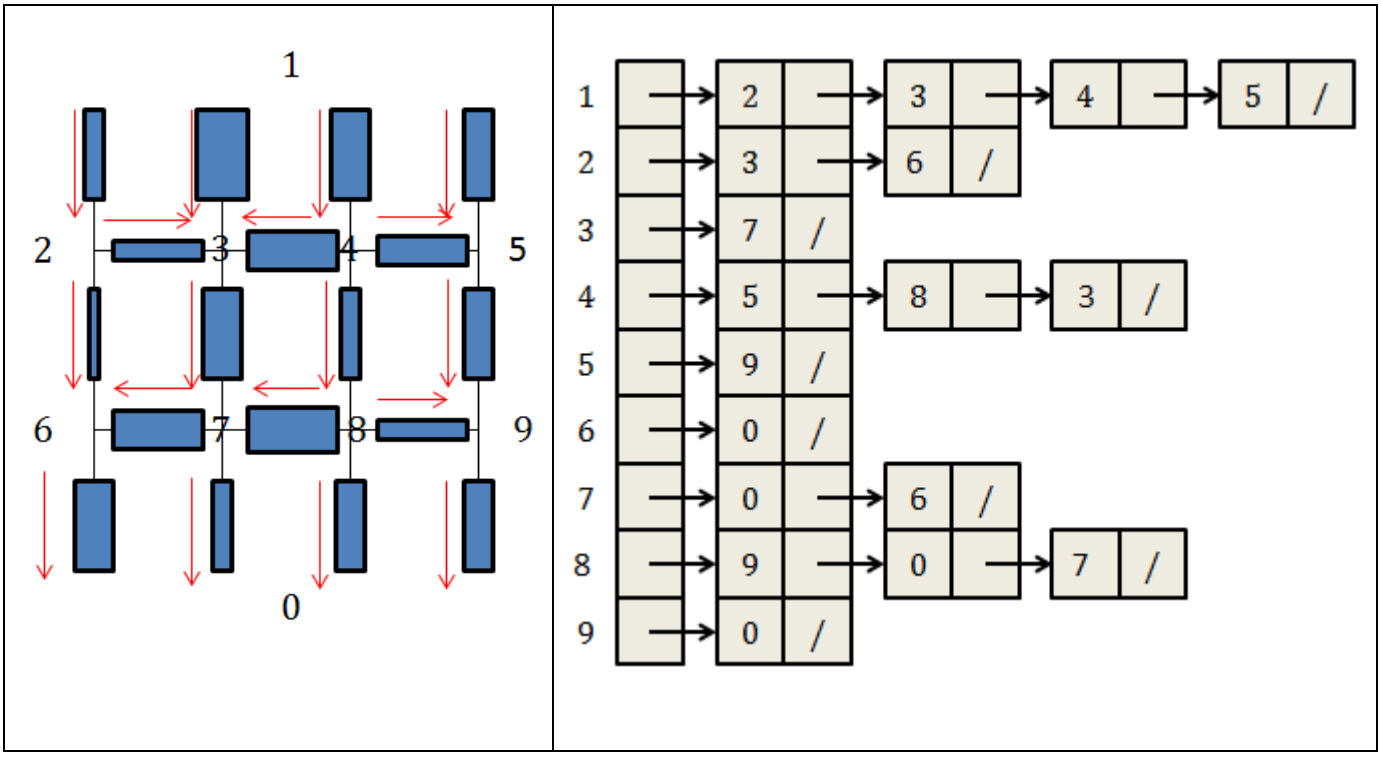


Figure 4.2: (a) Example directed network. (b) Its Adjacency list representation.

4.3 Fluid Arrangement in Tube

Consider a single tube in the simplified two-phase system. Let this tube connect to node i and node j. Albeit abstract, let this tube be initially filled with a single fluid, referring to it as the defending fluid, and allow it to have viscosity μ_{def}. After some time, assume that an invading fluid with viscosity μ_{inv} invades the tube with length d and radius r from node i. Figure 4.3 shows the fluid arrangement.

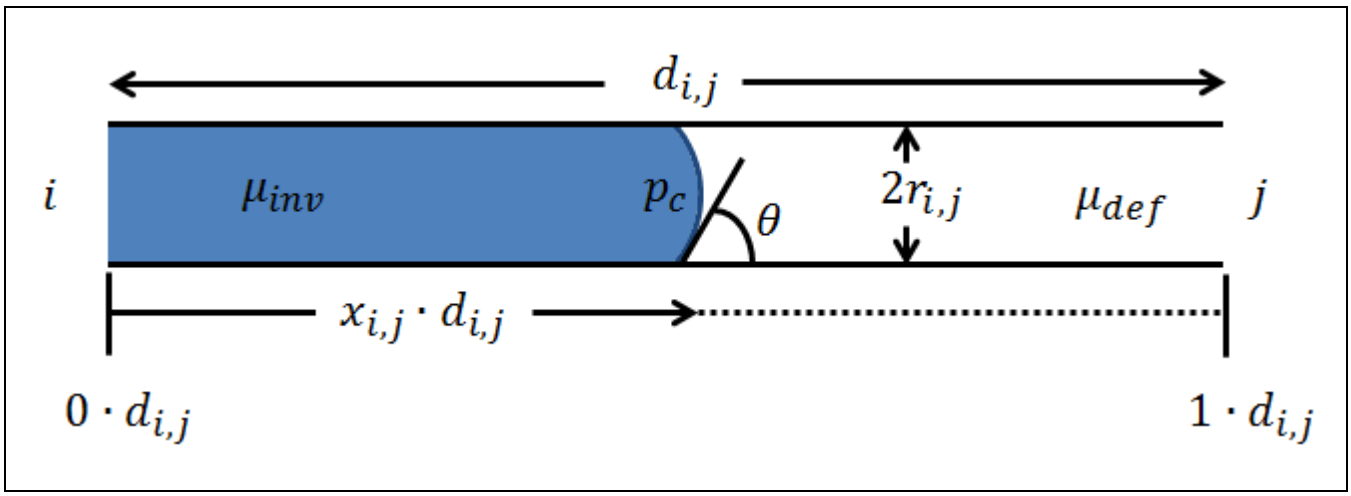


Figure 4.3: Abstract fluid arrangement in a tube.

As Figure 4.3 shows, variable $x_{i,j}$ represents a fraction of the length of the tube and follows the penetration of the invading fluid into the tube. Therefore, $x_{i,j}$ is between zero and one inclusive.

The conductance for a tube filled with a single fluid is visible in Equation 2.2. Because conductance is the reciprocal of resistance, the resistance is known from this expression. To derive the equivalent conductance of the tube shown in Figure 4.3, the tube is viewed as two tubes connected in series. In the fluid configuration shown, the resistor with viscosity μ_{inv} has a length of $x_{i,j}d$ and the resistor with viscosity μ_{def} has length $(1 - x_{i,j})d$. Recalling Equation 3.5, the equivalent resistance of the tube is

$$R_{i,j} = \frac{8\mu_{inv}x_{i,j}d_{i,j}}{\pi r_{i,j}^4} + \frac{8\mu_{def}(1 - x_{i,j})d}{\pi r_{i,j}^4}. \quad [4.1]$$

Therefore, the conductance of the tube, which is more amenable to MNA, is

$$G_{i,j} = \frac{\pi r_{i,j}^4}{8d[\mu_{inv}x_{i,j} + \mu_{def}(1 - x_{i,j})]}. \quad [4.2]$$

From the expression, the conductance of an edge is easily computed knowing the type of the defending fluid and penetration of the invading fluid into each tube. For example, given a non-wetting fluid with viscosity $1Pa \cdot s$ injected into a network initially filled with wetting fluid with viscosity $0.001Pa \cdot s$, if some instance of an edge class in the network at some time $t > 0$ has a defending type attribute of wetting and $x_{i,j}$ attribute of 0.2, it is known that the non-wetting fluid has invaded into the tube a fifth of the tube's length. In this example, it is also known that this particular edge object must use $\mu_{def} = 0.001Pa \cdot s$ and $\mu_{inv} = 1Pa \cdot s$ to compute its conductance. Therefore, to adequately track and compute the conductance for each tube, the edge class under the object-oriented programming paradigm must define defending type and $x_{i,j}$ attributes.

Note that in Equation 4.2, the term $\mu_{inv}x_{i,j} + \mu_{def}(1 - x_{i,j})$ is the effective viscosity, $\mu_{i,j_{eff}}$, of the tube. Making the substitution in the equation and noting that $\Delta P_{i,j} = P_i - P_j - p_{c_{i,j}}$, the volumetric flow rate of a two-phase tube is

$$q_{i,j} = \frac{\pi r_{i,j}^4}{8\mu_{i,j_{eff}} d_{i,j}} \Delta P_{i,j}. \qquad [4.3]$$

This expression is the Washburn equation. Because the meniscus only advances if $P_i - P_j > p_{c_{i,j}}$, the flow rate is algorithmically given by

$$q_{i,j} = \begin{cases} \frac{\pi r_{i,j}^4}{8\mu_{i,j_{eff}} d_{i,j}} \Delta P_{i,j}, & P_i - P_j > p_{c_{i,j}}. \\ 0, & otherwise \end{cases} \qquad [4.4]$$

As the next section discusses, to advance the flow pattern in time, the procedure is equivalent to solving Equation 4.4 for each time step.

4.4 Algorithm Overview

Menisci advancement is algorithmically challenging. At $t = 0$, the geometry and interconnection of the tubes, properties of the fluids, and sources of excitation are all known. However, nothing is known about the network at the next time step considering the inherent interdependencies

among the variables. Obviously, the pressure distribution and saturation mutually affect each other; the pressure differences across and flux through each tube affects the invasion of the non-wetting fluid into the network while concurrently, this invasion instantaneously affects the pressure differences across and flux through each tube as the conductance of each multiphase tube changes through this dynamic process. Because it is unknown how any of these parameters evolve by the next time step, an algorithm based on some sort of implicit method is not an option; an explicit approach is needed.

Adopting an explicit mindset, the conductance of each tube is first computed during a time step from the attributes of the nodes and edges. Using each conductance and sources of excitation, the pressure is then computed from MNA followed by the flux for each tube from $q_{i,j} = G_{i,j}\Delta P_{ij}$. At this point, the saturation is updated. While straightforward, the matter of choosing an appropriate time step and advancing the menisci remains.

4.5 Menisci Advancement in Tube

Reconsider Figure 4.3 with $x_{i,j} = 0$. Assuming the type attribute of node i differs from the defending fluid attribute of the tube and an adequate pressure difference exists to drive the invading fluid, the meniscus at $x_{i,j}$ needs to advance. Because the velocity of the flow in the tube is the volumetric flow rate divided by the cross-sectional area of the tube, the normalized velocity at some moment in time is given by

$$\frac{dx_{i,j}}{dt} = \frac{q_{i,j}}{\pi r_{i,j}^2 d_{i,j}}. \qquad [4.5]$$

For some iteration, the right-hand side of Equation 4.4 is a constant. Using a forward Euler discretization of order $O(\Delta t)$, the next position of $x_{i,j}$ is given by

$$\left(x_{i,j}\right)_{n+1} = \left(x_{i,j}\right)_n + \frac{\Delta t \cdot q_{i,j}}{\pi r_{i,j}^2 d_{i,j}}, \qquad [4.6]$$

where Δt is the step size (Sewell, 2005). Alternatively, using a central difference discretization of order $O((\Delta t)^2)$, the next position of $x_{i,j}$ is given by

$$\left(x_{i,j}\right)_{n+1} = \left(x_{i,j}\right)_{n-1} + \frac{2\Delta t \cdot q_{i,j}}{\pi r_{i,j}^2 d_{i,j}}. \quad [4.7]$$

Use of Equation 4.7 requires that the program keep a record of the previous position in addition to the current position and that Equation 4.6 be used for the first advancement of the meniscus. Although the results in this work are attained using only Equation 4.6, the previous position would simply be an additional attribute in the edge class and its use could easily be weaved into the algorithm.

4.6 An Appropriate Time Step

Because the pressure distribution and saturation mutually affect each other, an appropriate time stepping mechanism is needed to hopefully ensure a reliable flow pattern. Choosing a maximum normalized step length Δx_{max} a meniscus may travel during a time step, each multiphase tube has a time associated with it computed from either Equation 4.6 or Equation 4.7 depending on the chosen simulator. After cycling through each multiphase tube, the minimum of these times is taken to be the time step to advance the simulation. For a simulator that uses only Equation 4.6 to advance the network, the time step is chosen using

$$\Delta t = \min_{i,j}\left(\frac{\Delta x_{max}\pi r^2}{q_{i,j}}, \frac{\left(1 - x_{i,j}\right)\pi r^2}{q_{i,j}}\right), \quad [4.8]$$

which ensures that no tube advances beyond its physical capacity. A similar expression follows for a simulator which uses Equation 4.7.

4.7 Flow Rules

At some stage during a simulation, the network is composed of nodes with different type attributes and tubes with different defending type and $x_{i,j}$ attributes. To make sense of the fluid arrangements and unambiguously advance the fluids, a fluid organization technique is used prior to

advancing each meniscus to ensure that the type attribute of a source node in an adjacency list matches the fluid represented by the $x_{i,j}$ attributes of its outgoing tubes. Consequentially, the type of a source node in the structure must be different from the defending type attribute of each of its outgoing tubes.

At the beginning of an iteration, the conductance and nodal pressures are computed followed by the flux for each tube. Assuming a directed graph implementation, if the flux for any tube in an outgoing tube is negative, the direction has changed from the previous iteration and the directed graph must be corrected such that the tube in the data structure is outgoing with the correct source node and destination node. For such an occurrence, the element i from the linked list of $Adj\ [j]$ is removed and inserted into the linked list of $Adj[i]$. In addition to storing the magnitude of the flux in the moved link's flux attribute field, the defending type and $x_{i,j}$ attributes of the edge potentially change. For a source node whose type attribute differs from the defending type attribute of the edge, the defending type and $x_{i,j}$ attributes are unchanged. However, for a source node whose type attribute matches the defending type attribute of the edge, $x_{i,j} = 1 - x_{i,j}$ and the defending type attributed is changed to the other fluid to effectively rearrange the fluids in the tube.

Although not always a point of agreement among programmers to allow a function to perform more than one task, the fluid organization technique and computation of a time step can be coded into a single method. In this fashion, as the program examines a tube in the adjacency list, its fluids are rearranged if need be and if it contains both phases and has a non-zero flux, the time step to advance it is computed and compared against the known smallest of such times. To identify if a tube is two-phase, it is simplest to identify that the tube is not single-phase. According to the programming view of the network, a tube is single-phase when the type of the source node and the defending type of the tube match and $x_{i,j} = 0$.

After properly configuring the fluids and arriving at a time step, the menisci are advanced. To complete this operation, the program loops through the adjacency list twice. During the first pass, for each outgoing tube whose defending fluid type differs from its corresponding source node, its meniscus is advanced according to Equation 4.6 or Equation 4.7 depending on the chosen simulator. During the second pass, the program updates the types of the nodes and defending types of the tubes if need be by

checking for occurrences where the penetration attribute is $x_{i,j} = 1$. For such instances, if the defending fluid of the tube is wetting, the defending type of the tube and destination node associated with the edge both change to non-wetting. If the defending fluid of the tube is non-wetting, the defending fluid of the tube changes to wetting as before, but additional checks are needed to determine if the destination node associated with the edge changes type. This interpretation gives preference to the non-wetting fluid to ensure the non-wetting fluid does not become disconnected. It also attempts to ensure that backflow occurs in all previous outgoing tubes before the node is re-declared. Before describing the check to change a non-wetting node back to a wetting node, it is worth noting that it is incorrect to combine both passes of the adjacency list to advance the fluids into one pass because it then becomes possible for a tube to fill, the destination node of the tube to change type, and that node acting as a source node further down the adjacency list to begin to fill its outgoing tubes. It is also worth noting that for the occurrences of $x_{i,j} = 1$, after the defending type of the tube and potentially its destination node's type change, the penetration attribute is reset such that $x_{i,j} = 0$ to reflect that the tube is now single-phase.

Checking a node that is potentially no longer non-wetting is actually more computationally expensive using a directed graph implementation than its undirected counterpart because all tubes associated with the node need to be checked for certain conditions and not just the outgoing tubes. However, such checks are assumed to occur infrequently in a drainage simulation. In order for the node to be declared wetting, all tubes must meet the following conditions: (1) for all incoming tubes, the penetration attribute must be $x_{i,j} = 1$ for tubes with defending type non-wetting and $x_{i,j} < 1$ for tubes with defending type wetting, and (2) for all outgoing tubes, the penetration attribute must be $x_{i,j} > 0$ for tubes with defending type non-wetting and $x_{i,j} = 0$ for tubes with defending type wetting. If any of the conditions fail, the node is not declared wetting.

Unless one of three stopping condition arrives, the current iteration is complete and the program moves onto the next iteration. Despite lacking a physical reason to assume the non-wetting fluid does not further its invasion in other areas of the network, one stopping condition is taken when breakthrough, or when the non-wetting fluid first reaches the outlet node, occurs. Another stopping condition occurs when the program reaches the end of its iteration count. The final terminating

condition can actually occur before the fluid advances during the current iteration and actually arises during the computation of the current iteration's time step. Before looping through the adjacency list to find the time step, the known smallest time step is taken to be infinity. When the program loops over the first multiphase tube it encounters, the time step equals the time associated with that tube provided the flux is greater than zero. Therefore, an insufficient pressure difference can cause the program to never find a multiphase tube with nonzero flux and return the initial infinite time step. Because an infinite time step is not physically useable, the program terminates. This stopping condition should only occur when a network is specified with only pressure sources.

4.8 Miscellanea

The content within this section is devoted to aspects of the model that do not technically contribute to the displacement of the fluids, but are essential to the simulator. These aspects include the contents of the netlist and visualization of the network.

4.8.1 Netlist

The netlist is a text file that completely describes the network. Although much more difficult to read, it conveys the same information as a flow regime plot.

The netlists are all generated by an external program. The algorithm this program follows depends upon the specific network and for the square lattice of tubes reported upon, is relatively straightforward to form. The algorithm first writes the flow time, number of injection sources, number of pressure sources, number of non-reference nodes, number of edges, μ_1, μ_2, and γ. Although there is some freedom in the order in which these data are written to file, this first set of data in the netlist allows the simulator to initialize the network when read from the netlist file. The netlist generating algorithm next writes the source information to file. As before, the order in which these sources are written to file is arbitrary as long as the simulator and netlist generator follow the same order; the implementation of this algorithm writes the injection sources first followed by the pressure sources. In this set of data, a source is completely described by the nodes its terminals connect and its nominal value. Node information is then written to the file. Such information includes node number and type signifying if wetting or non-wetting fluid are at those locations. Finally, edge information is written to file. Such

information includes the source node and destination node of the tube, defending fluid type, penetration $x_{i,j}$, radius, length of tube, coordinates of the source node, and coordinates of the destination node. While arguably counterintuitive to not write the coordinates of the nodes with the node information of the file, it is beneficial to include the coordinates as edge information. This choice allows the simulator to view source and sink nodes as being composed of nodes of their own. In this fashion, while electrically a single entity, a node that supplies or consumes material is distributed among several nodes. Because this point of view allows the simulator to map more than one coordinate to these nodes, flow regime patterns can be strategically arranged when drawn such that the tubes are not cluttered around a single point. Therefore, to accurately convey the coordinates to the proper tube and avoid numbering the same node with more than a single number, this information is included with the edge information in the netlist file.

4.8.2 Visualization

Generation of flow regime patterns involves basic computer graphics. Although the tubes are geometrically given as cylinders, each edge is drawn as a rectangle to depict the invasion from its so-called side view. Therefore, no more than basic matrix algebra and a polygon drawing routine are needed to replicate this aspect of the model. Because this drawing routine needs the coordinates of its vertices, the content within this section outlines how to attain these values.

Consider a single tube. Let node i and node j of this tube each have (x, y) coordinates in the standard Cartesian coordinate system such that the distance between these points is $L_{i,j}$ where $L_{i,j} \geq d_{i,j}$. Let the axis of the "cylinder" be along the straight-line path between the nodes where the centers of both the path and tube are aligned. As before, let this tube be initially filled with a single fluid with viscosity μ_{def} and after some time, be invaded by a fluid with viscosity μ_{inv} from node i. Figure 4.4 depicts this situation.

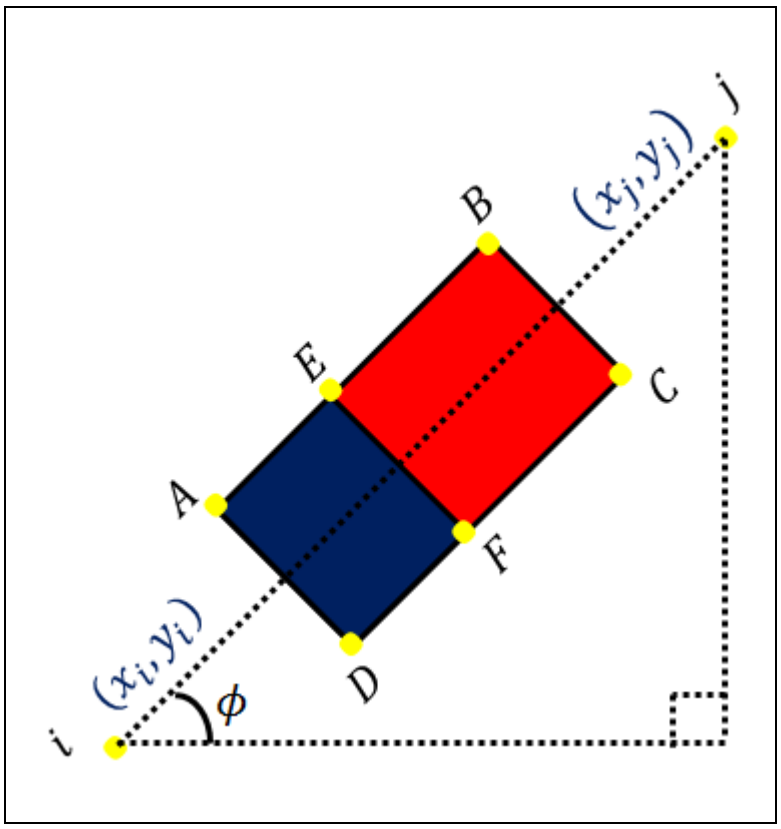


Figure 4.4: Position of a general tube.

As the figure shows, the slant of this general tube is conveniently described by the angle ϕ, measured counterclockwise from the horizontal reference through node i. Because the tube's orientation between its nodes has been defined and its geometrical attributes and penetration attribute are known, the six points which outline the tube and its invasion are calculable. Once these points are computed, the polygon drawing routine is invoked twice to draw two rectangles depicting the tube. For example, the implementation might draw rectangle $ABCD$ followed by rectangle $AEFD$, thereby overlapping some of the larger rectangle's area. Alternatively, drawing rectangles $AEFD$ and $EBCF$ without any overlap would achieve the same result. For the undirected version of the adjacency list, drawing a tube requires two passes of the data structure. During the first pass, rectangle $ABCD$ is drawn for each tube with a color to depict its defending fluid. During the second pass, rectangle $AEFD$ is drawn for each tube with a color to depict the invading fluid. Note that the defending type attribute of a tube is used to signify to the polygon drawing routine which colors to fill the rectangles representing the tubes.

A mapping procedure allows the simulator to arrive at the six points describing the tube and its invasion. In this fashion, the six points are determined in some coordinate system and through a set of transforms, are mapped to the correct locations. While other starting coordinate systems are possible, Figure 4.5 shows the starting coordinate system for each tube the implementation uses.

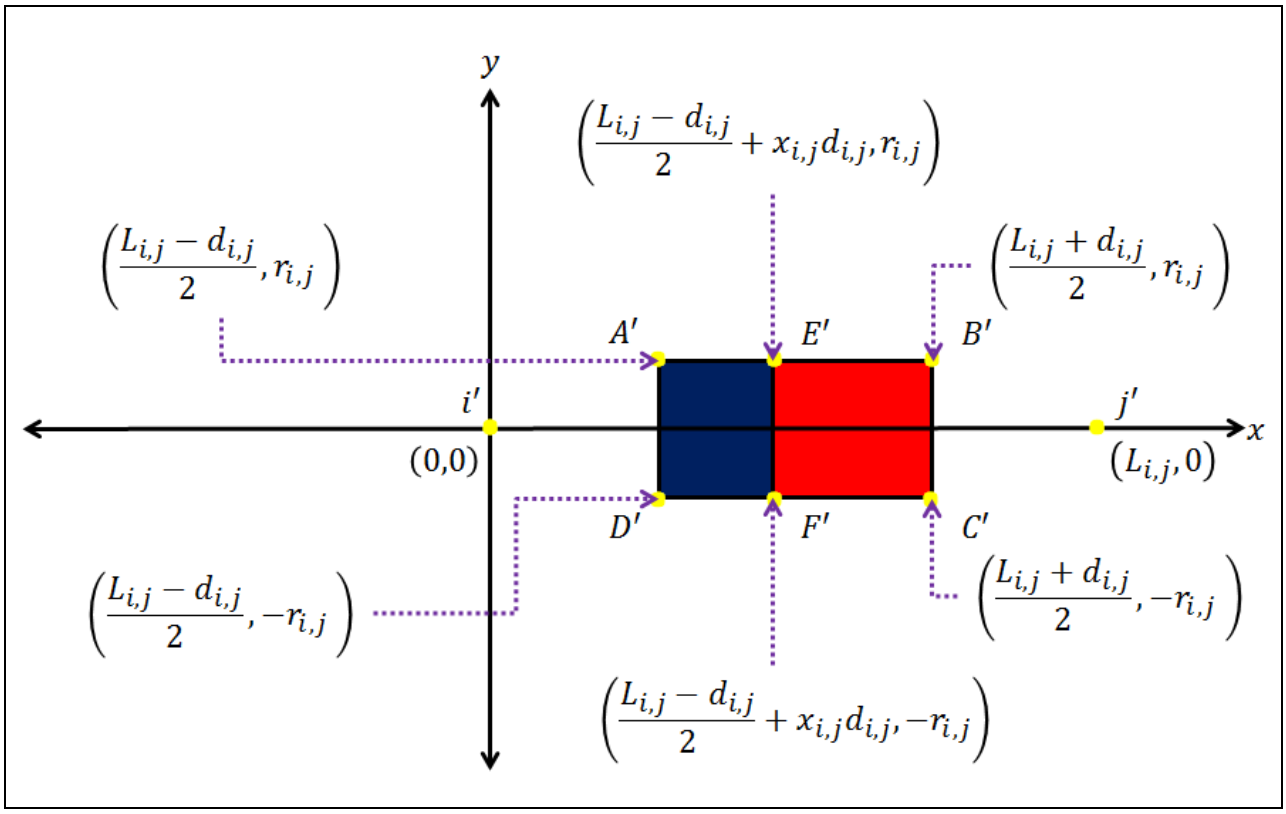


Figure 4.5: Starting position of a general tube.

Notice that each point in the figure is now labeled with an apostrophe. Therefore, a tube is correctly mapped by transforming each point $z' \rightarrow z$ where $z = \{A, B, C, D, E, F, i, j\}$. Because the coordinates of i and j are known and the coordinates of i' and j' are defined in the figure, the operations needed to arrive at $i' \rightarrow i$ and $j' \rightarrow j$ are completely determinable.

The algorithm starts by determining ϕ in Figure 4.4. Because the angle is measured counterclockwise from the horizontal reference through node i, conditional statements are needed to ensure that the correct value of the angle is returned. Let $u = x_j - x_i$ and $v = y_j - y_i$. If $u > 0$ and

$v \geq 0,\ \phi = \tan^{-1}\frac{v}{u}$. If $u < 0$ and $v \geq 0,\ \phi = 180° - \tan^{-1}\frac{v}{u}$. If $u < 0$ and $v < 0,\ \phi = 180° + \tan^{-1}\frac{v}{u}$. If $u > 0$ and $v < 0$, $\phi = 360° - \tan^{-1}\frac{v}{u}$. If $u = 0$ and $v > 0$, $\phi = 90°$. Finally, if $u = 0$ and $v < 0$, $\phi = 270°$.

After ϕ is determined, the tube in Figure 4.5 is rotated. Let this transformation be signified as $i' \rightarrow i''$ and $j' \rightarrow j''$ since those nodes are not at the desired locations after rotation. To actually rotate the tube, a matrix vector multiplication is carried out. If i' is at $(x_{i'}, y_{j'})$ and j' is at $(x_{j'}, y_{j'})$, then the coordinates for i'' and j'' are determined from

$$\begin{bmatrix} x_{i''} \\ y_{i''} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i'} \\ y_{i'} \\ 1 \end{bmatrix}, \qquad [4.9]$$

and

$$\begin{bmatrix} x_{j''} \\ y_{j''} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{j'} \\ y_{j'} \\ 1 \end{bmatrix}, \qquad [4.10]$$

where i'' is at $(x_{i''}, y_{i''})$ and j'' is at $(x_{j''}, y_{j''})$. Notice that matrix is 3×3 despite the two-dimensional coordinate system. Setting up the matrix in this fashion allows the transform to be carried out with matrix multiplication operations.

Finally, a translation operation effectively slides the rotated nodes to the locations specified in the netlist. Let this transformation be signified as $i'' \rightarrow i$ and $j'' \rightarrow j$. Because i' is initially set to the origin in Figure 4.5 (and remains so when transformed to i''), the node arrives at its proper location by adding (x_i, y_i) to $(x_{i''}, y_{i''})$. Similarly, j'' is transformed to j by adding (x_i, y_i) to $(x_{j''}, y_{j''})$. This operation is denoted by

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_i \\ 0 & 1 & y_i \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i''} \\ y_{i''} \\ 1 \end{bmatrix}, \qquad [4.11]$$

and

$$\begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_i \\ 0 & 1 & y_i \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{j''} \\ y_{j''} \\ 1 \end{bmatrix}. \quad [4.12]$$

In summary, the algorithm performs operations to transform $i' \rightarrow i'' \rightarrow i$ and $j' \rightarrow j'' \rightarrow j$. Let the matrix M equal the product of the two transformation matrices such that

$$\mathrm{M} = \begin{bmatrix} 1 & 0 & x_i \\ 0 & 1 & y_i \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad [4.13]$$

Therefore, the coordinates of i' and j' are transformed to the coordinates of i and j by

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = M \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad [4.14]$$

and

$$\begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} = M \begin{bmatrix} L_{i,j} \\ 0 \\ 1 \end{bmatrix}. \quad [4.15]$$

Although there is no need to transform i' and j' to i and j respectively, each coordinate for z is computed by

$$\begin{bmatrix} x_z \\ y_z \\ 1 \end{bmatrix} = M \begin{bmatrix} x_{z'} \\ y_{z'} \\ 1 \end{bmatrix}, \quad [4.16]$$

where its coordinate in Figure 4.4 is (x_z, y_z) and its starting coordinate z' as given in Figure 4.5 is $(x_{z'}, y_{z'})$. Because M is fixed for a general tube throughout the simulation, M is computed once and saved as an attribute in the edge class. Additionally, to properly group the six points with the correct tube, the coordinates for A, B, C, D, E, and F are designated as attributes in the edge class. Because the

points A, B, C, and D in Figure 4.4 never change, these points are also computed and saved once. Only points E and F are updated if more than a single flow regime is plotted for a network at multiple time instances.

Chapter 5: Results

5.1 Introduction

Because modified nodal analysis forms equations which essentially state that current does not accumulate at nodes and that the difference in pressure between two nodes with a connecting pressure source is the nominal pressure for that source, this aspect of the model is not the subject of these tests; the matrix equation is axiomatic. However, the premise of a tube being analogous to a resistor and the algorithmic flow rules are the aspects of the model under scrutiny.

Because the networks are not tailored to model a specific situation with known experimental data, the simulator outlined in the preceding chapters attempts to qualitatively match the flow patterns described in the Lenormand diagram. For a set of tests, a rectangular grid of cylindrical tubes initially filled with wetting fluid is generated. The top row of tubes serves as the inlet for the invading, non-wetting fluid while the bottom row serves as the outlet. The lengths of each tube are all the same while the tubes are assigned radii according to some probability distribution function; the lognormal, truncated normal and uniform distributions are considered to ensure that all radii are positive. Each of these networks is arbitrarily 88 by 88 with the length of its tubes being $d_{i,j} = 1mm$. Additionally, the fluids exhibit perfect wetting such that $\cos\theta = 1$.

In addition to the flow regime pattern, each simulation returns four plots to include the inlet pressure, saturation, average depth of penetration, and standard deviation of the average depth of penetration as functions of the flow time. The inlet pressure, measured relative to the outlet, provides insight into the evolution of the overall resistance of the network by injecting the foreign fluid. An increasing pressure with time signifies that the network becomes more resistive because more pressure is needed to maintain the desired injection rate. Similarly, a decreasing inlet pressure signifies a network with diminishing resistance. The saturation, or the fraction of non-wetting fluid to the sum of tube volumes, indicates how well the invading fluid displaces the valuable native fluid. Ideally, this value should be as close to one as possible. While not reported as distances in this report, the average depth of penetration and standard deviation of average depth of penetration provides insight into the front of the displacement. An average depth of penetration that increases to the maximum depth over time with a

small standard deviation signifies stable displacement while an average depth of penetration less than maximum and a large standard deviation signifies fingering.

The depth of penetration plots are not reported as distances due to the setup of the tubes. As supported by the description of the algorithm in the previous chapters, this simulator does not concern itself with where the tubes are drawn but rather how they are connected to one another. Because the physical position of the tubes does not hold meaning for the fluid displacements within the system, depth of penetration measurements are performed indirectly. Because of the regularity of the networks considered, these measurements are taken by examining the type attributes at the nodes in a column. Starting at an outlet node and moving up to an inlet node in the same column, the type of each node is examined. When a non-wetting node is located, the number of nodes between the inlet node in its column and that node is counted. This value represents the depth of penetration for that column. The average depth of penetration is computed as the average of such node counts while the standard deviation is computed using a similar approach after the average depth of penetration has been computed.

5.2 Lognormal Network

Figure 5.1 shows a lognormal distribution for the radii of each network shown in this section.

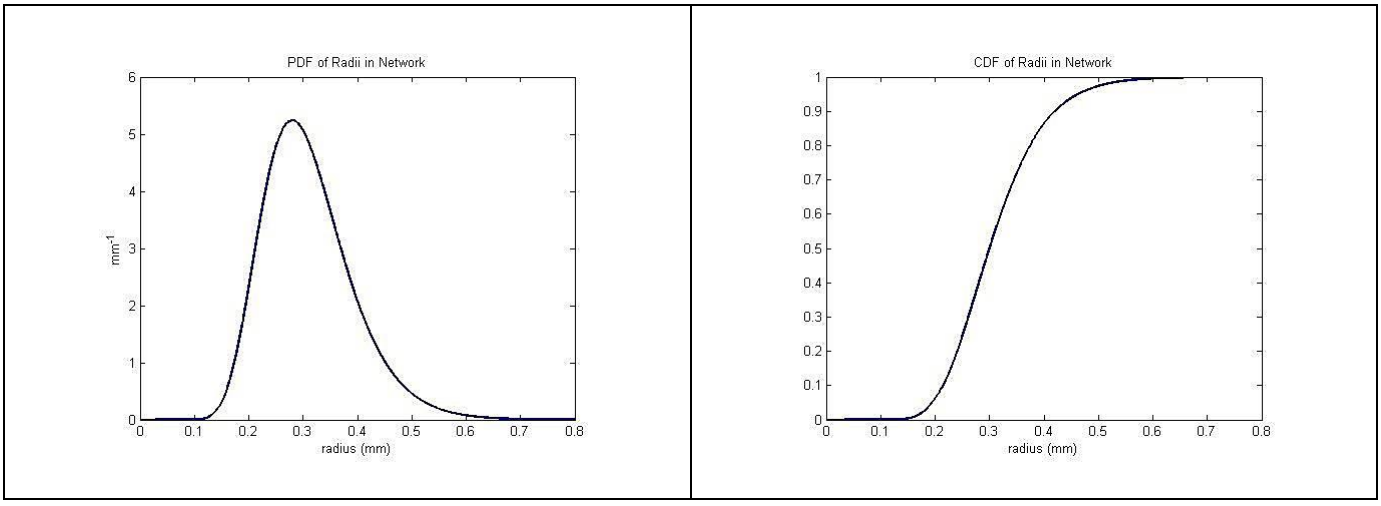


Figure 5.1: Lognormal PDF and CDF for radii in network.

5.2.1 Stable Displacement Regime

The example stable displacement regime is given for $(\log M, \log C_a) = (4, 6)$. To attain these values, the simulations use $\mu_1 = 0.0001$, $\mu_2 = 1$, $\gamma = 0.03$, $\Sigma = 2.6525 \times 10^{-5}$ and $Q = 0.79575$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.2 while Figure 5.3 through Figure 5.6 shows the four additional plots of interest.

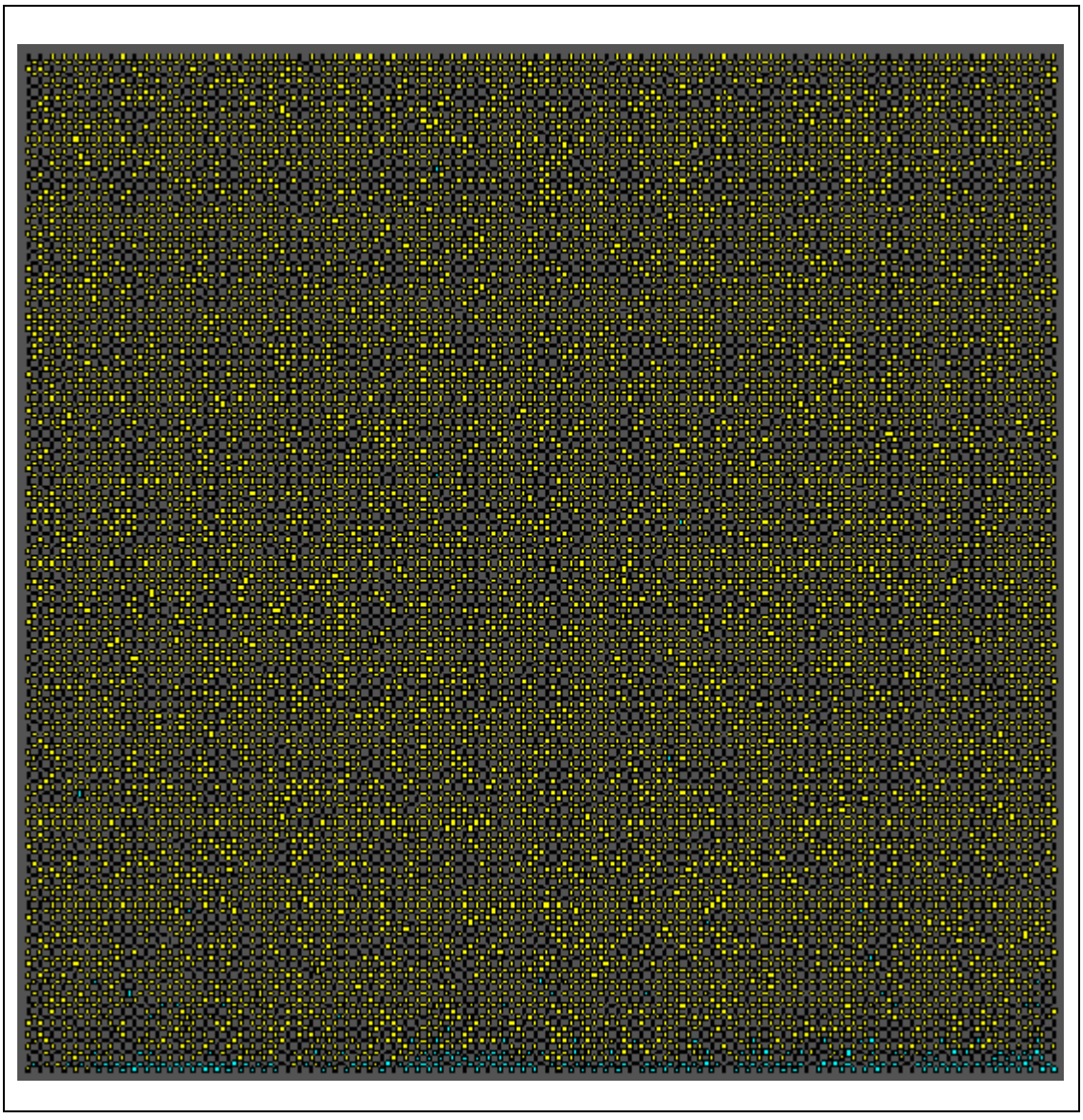


Figure 5.2: Stable displacement in lognormal network.

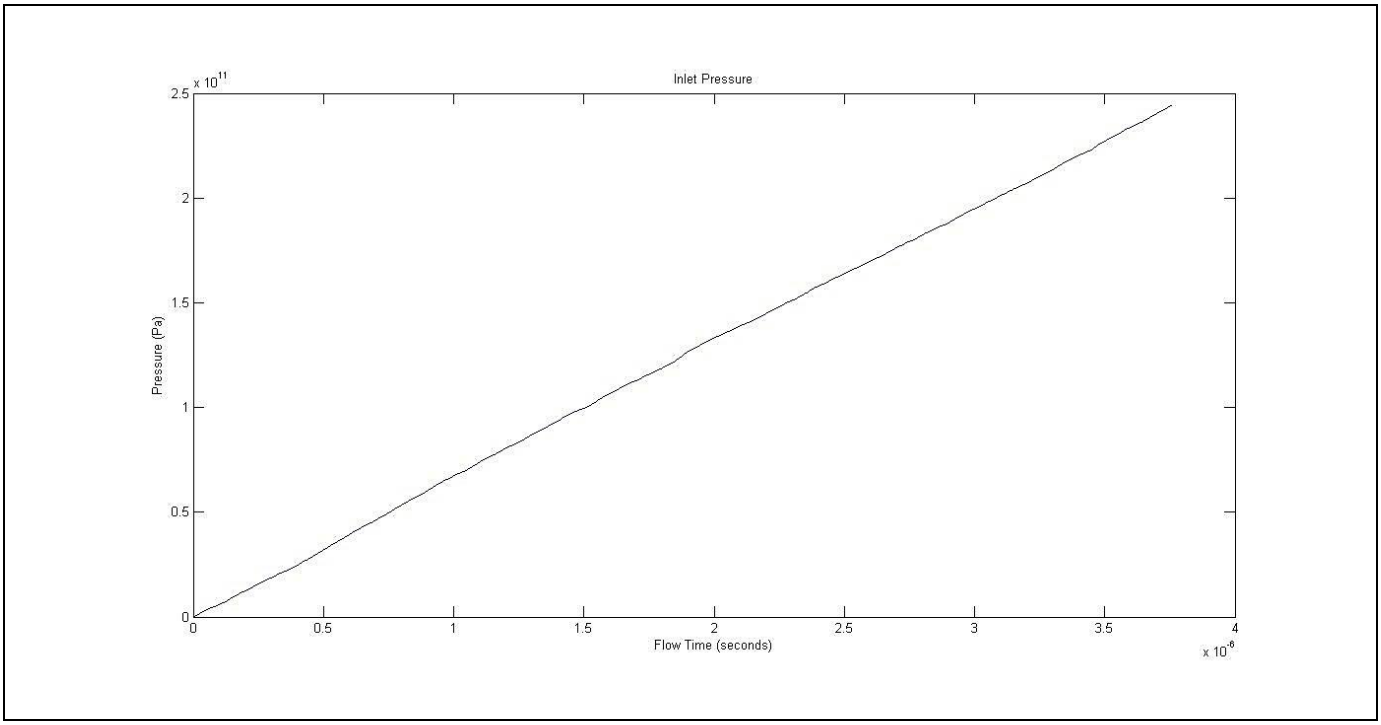


Figure 5.3: Inlet pressure for stable lognormal network.

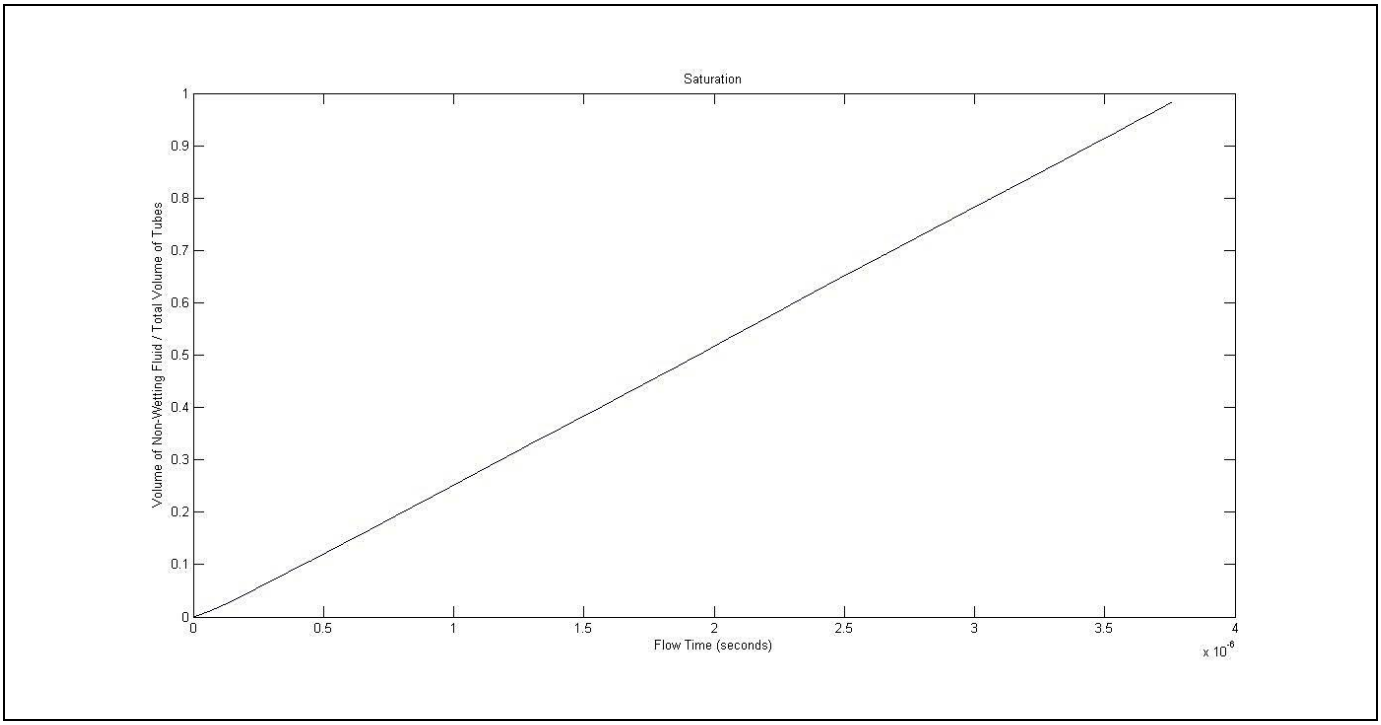


Figure 5.4: Saturation for stable lognormal network.

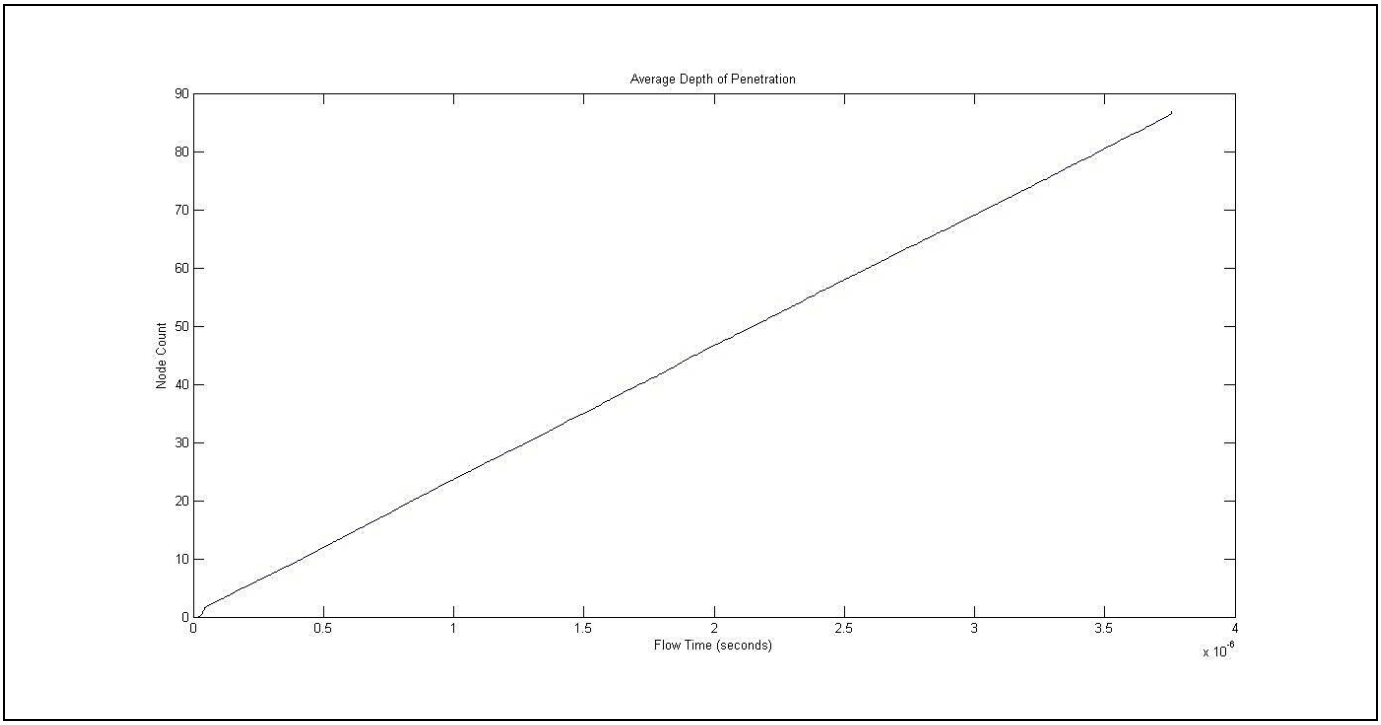


Figure 5.5: Average depth of penetration for stable lognormal network.

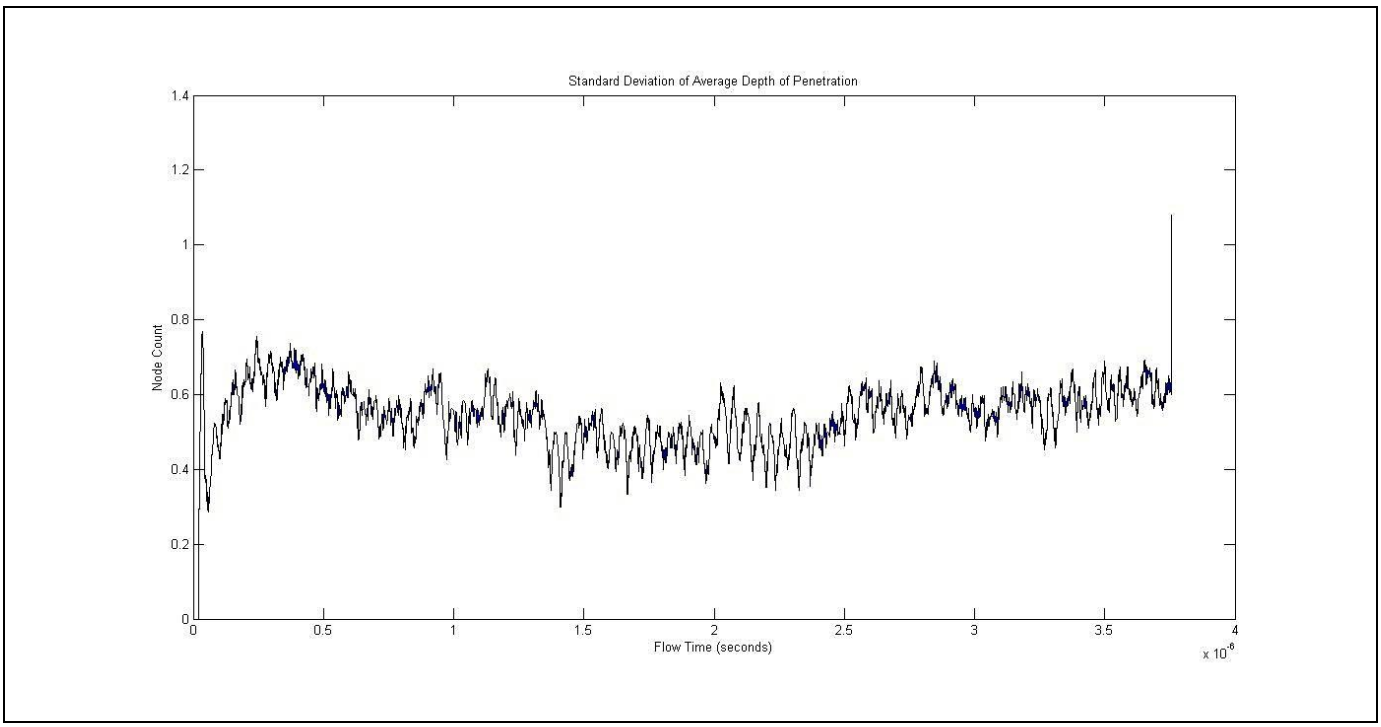


Figure 5.6: Standard deviation of average depth of penetration for stable lognormal network.

5.2.2 Viscous Fingering Regimes

For the viscous fingering regime in the Lenormand diagram, three flow regime plots are shown. These plots, generated with a different seed for the lognormal random number generator, show qualitatively the same information.

The first network that exhibits viscous fingering is given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations use $\mu_1 = 1$, $\mu_2 = 10^{-6}$, $\gamma = 0.03$, $\Sigma = 2.6525 \times 10^{-5}$ and $Q = 795750$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.7.

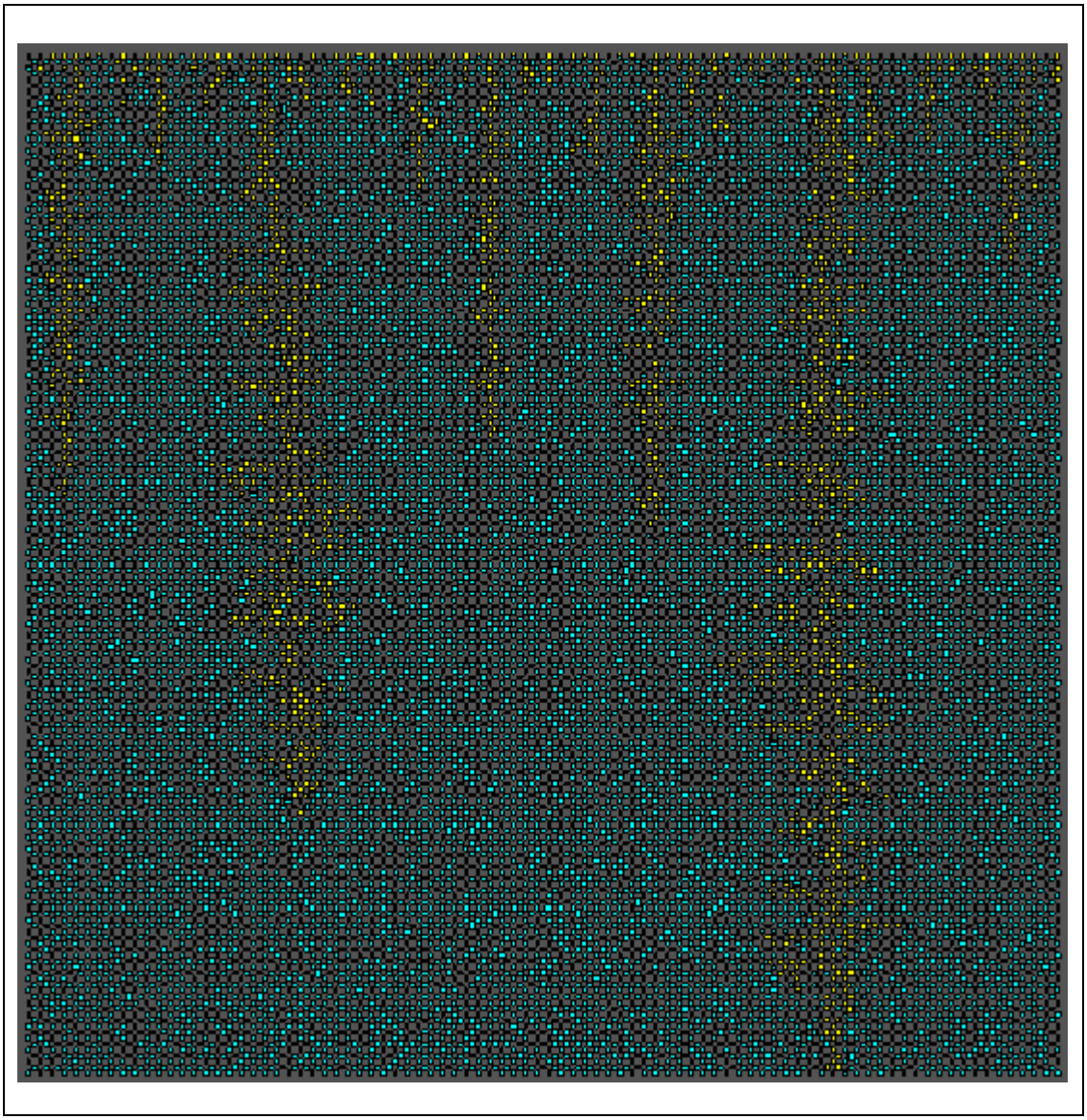


Figure 5.7: Viscous displacement in first lognormal network.

The second network that exhibits viscous fingering is also given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations now use $\Sigma = 2.5270 \times 10^{-5}$ and $Q = 758100$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.8.

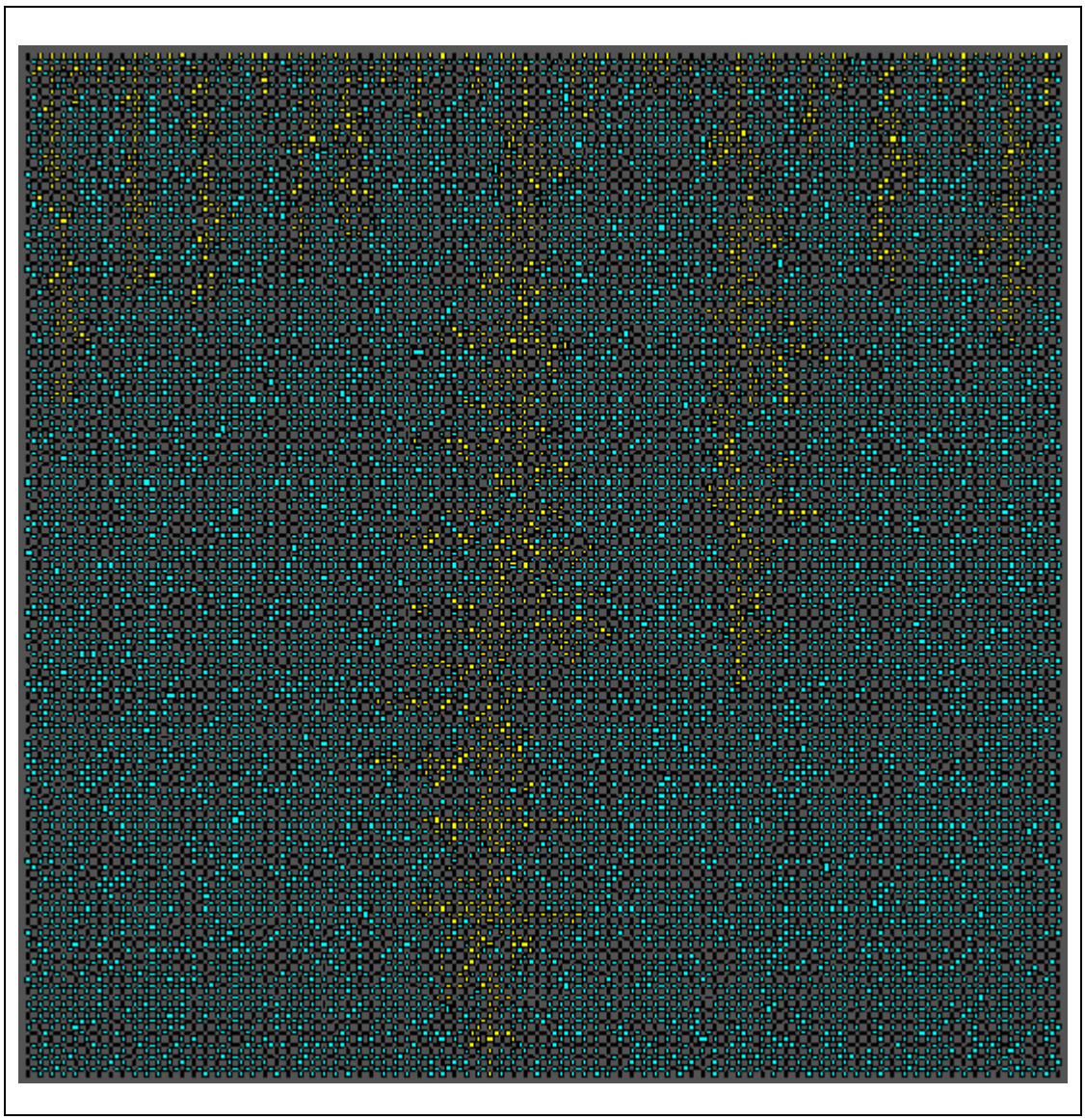


Figure 5.8: Viscous displacement in second lognormal network.

The third network that exhibits viscous fingering is also given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations now use $\Sigma = 2.9815 \times 10^{-5}$ and $Q = 894450$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.9.

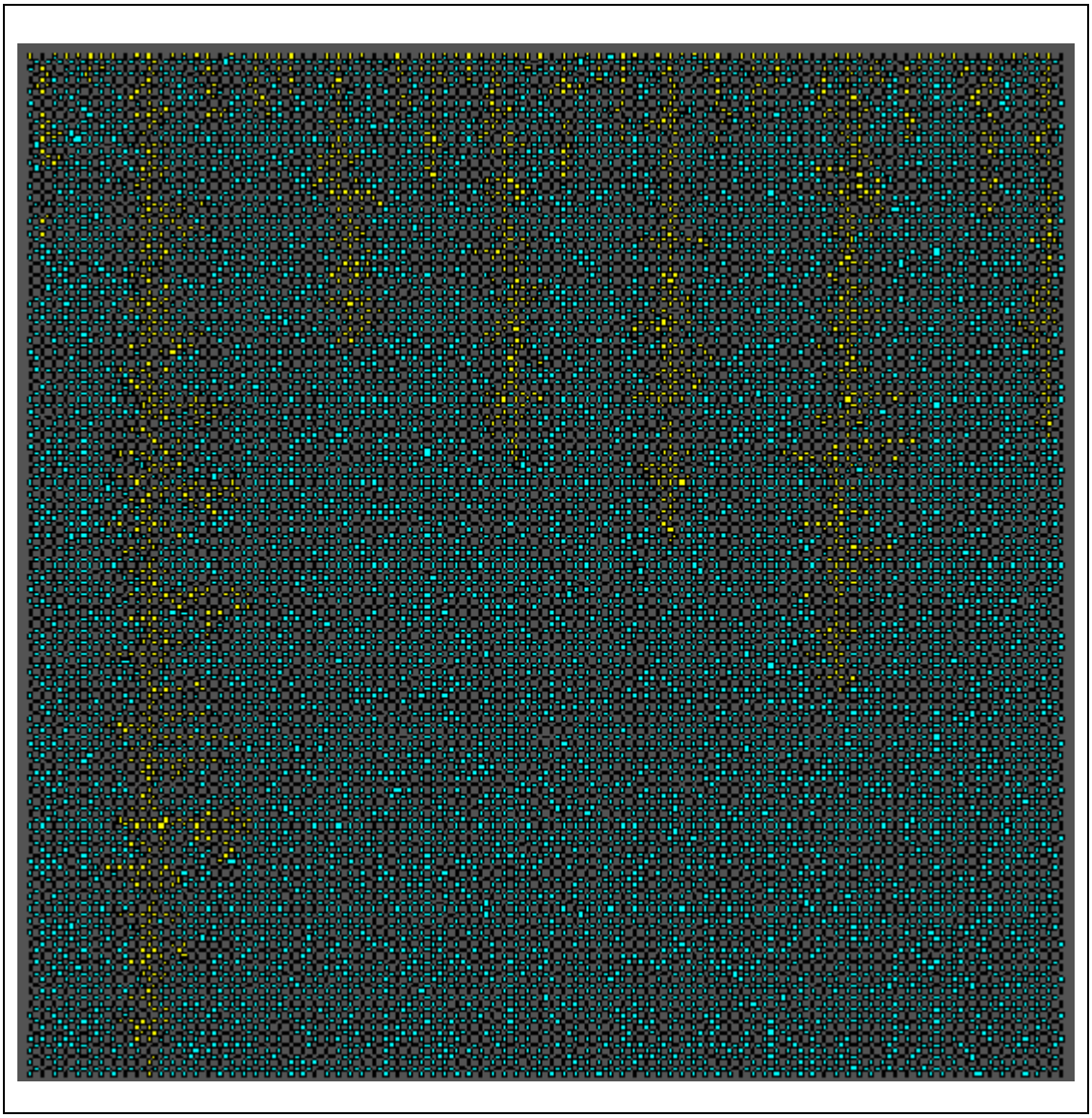


Figure 5.9: Viscous displacement in third lognormal network.

For the sake of comparison, Figure 5.10 through Figure 5.13 shows the four additional plots for each viscous network on the same coordinate planes. As these plots show and the flow regimes support, there is qualitatively no significant difference caused by the seeding in the random number generator.

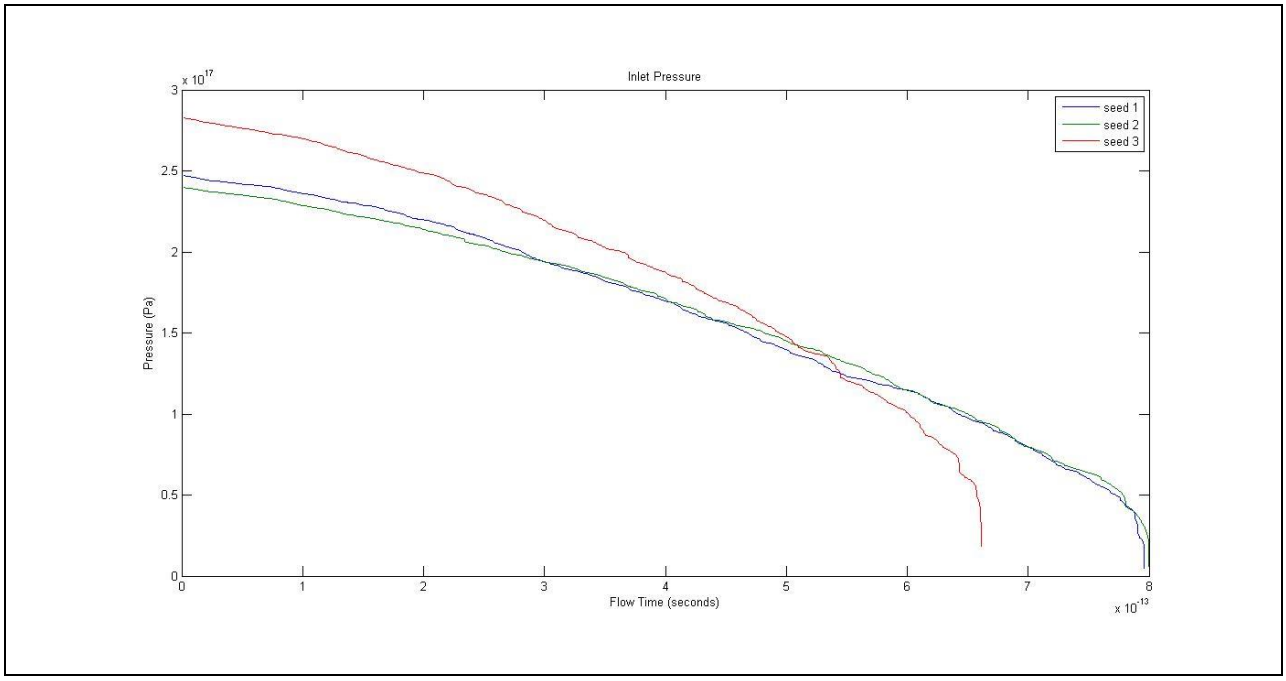


Figure 5.10: Inlet pressure for each viscous lognormal network.

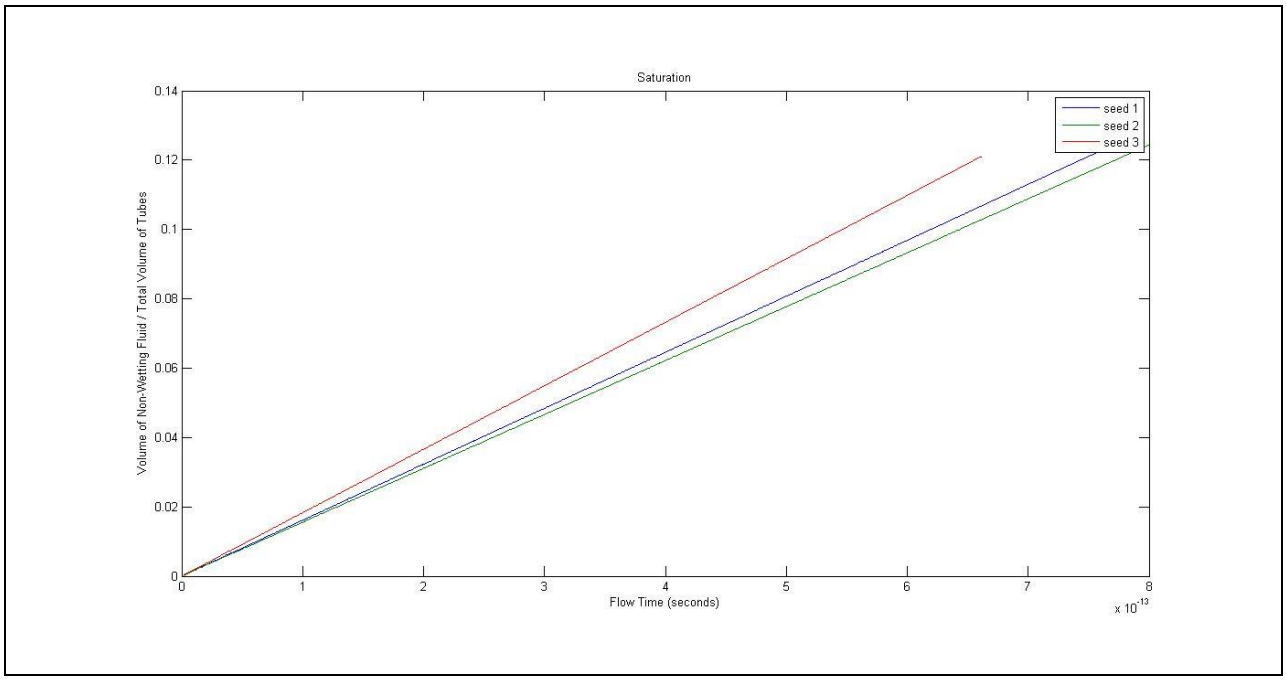


Figure 5.11: Saturation for each viscous lognormal network.

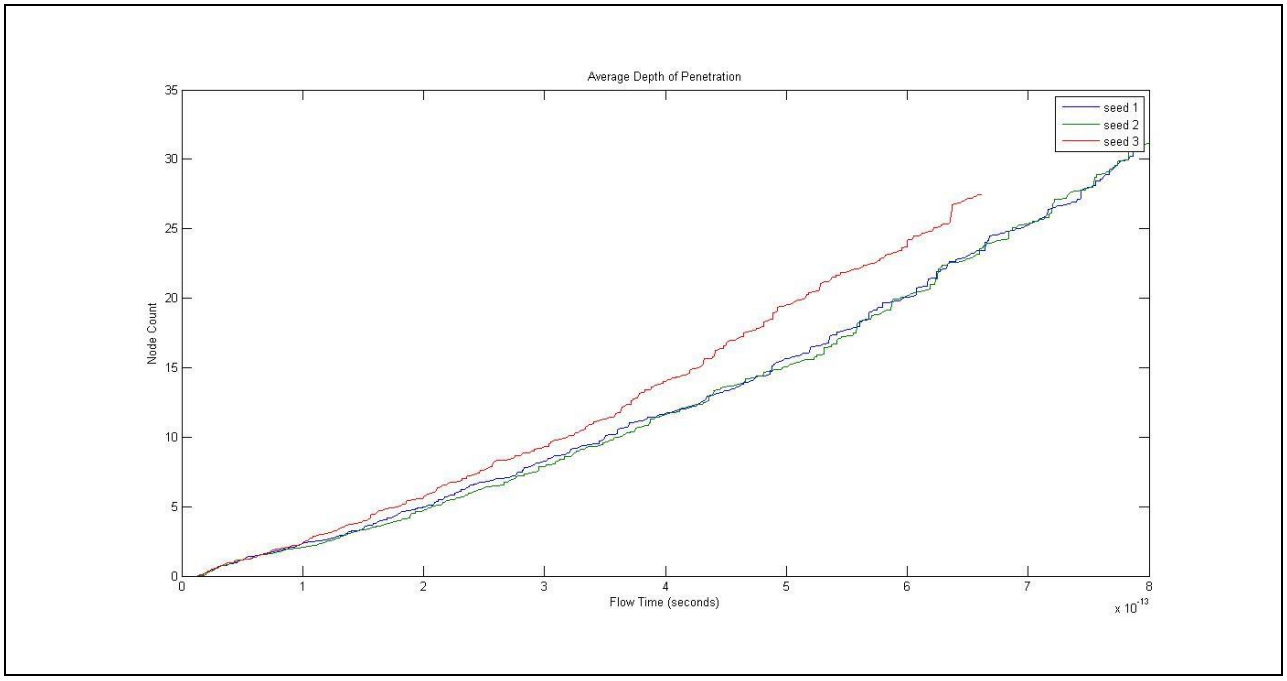


Figure 5.12: Average depth of penetration for each viscous lognormal network.

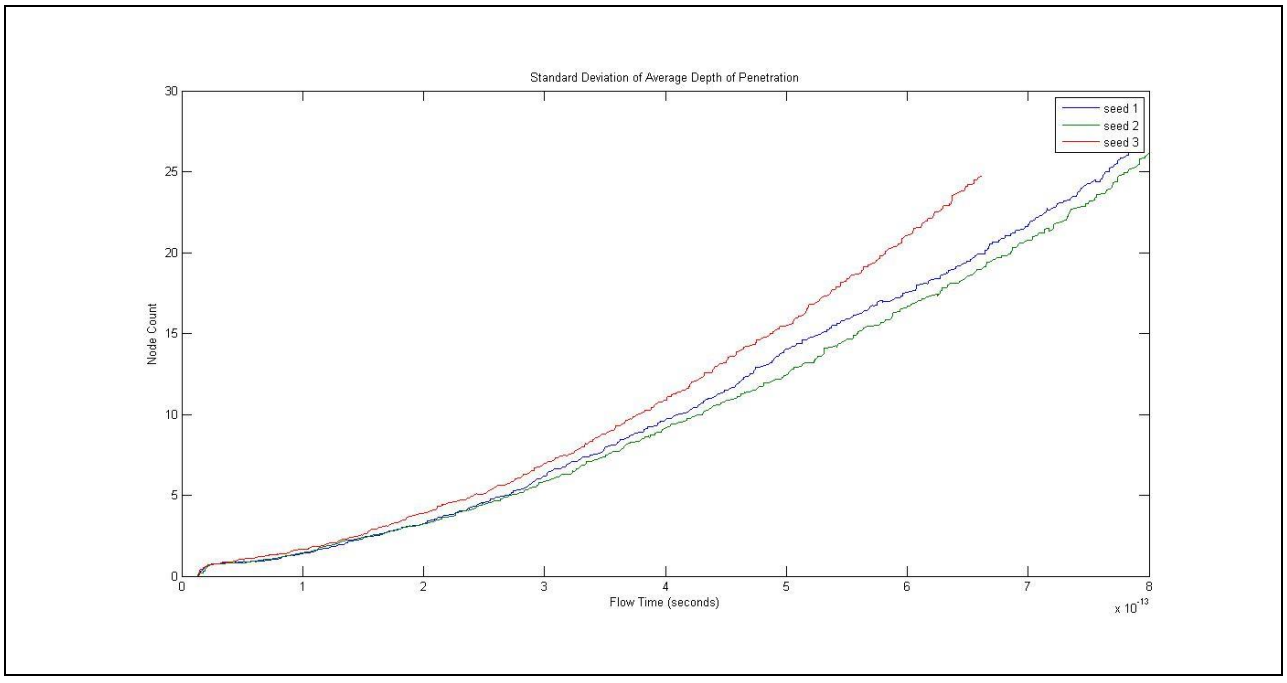


Figure 5.13: Standard deviation of average depth of penetration for each viscous lognormal network.

5.3 Truncated Normal Network

Figure 5.14 shows a truncated normal distribution for the radii of each network shown in this section.

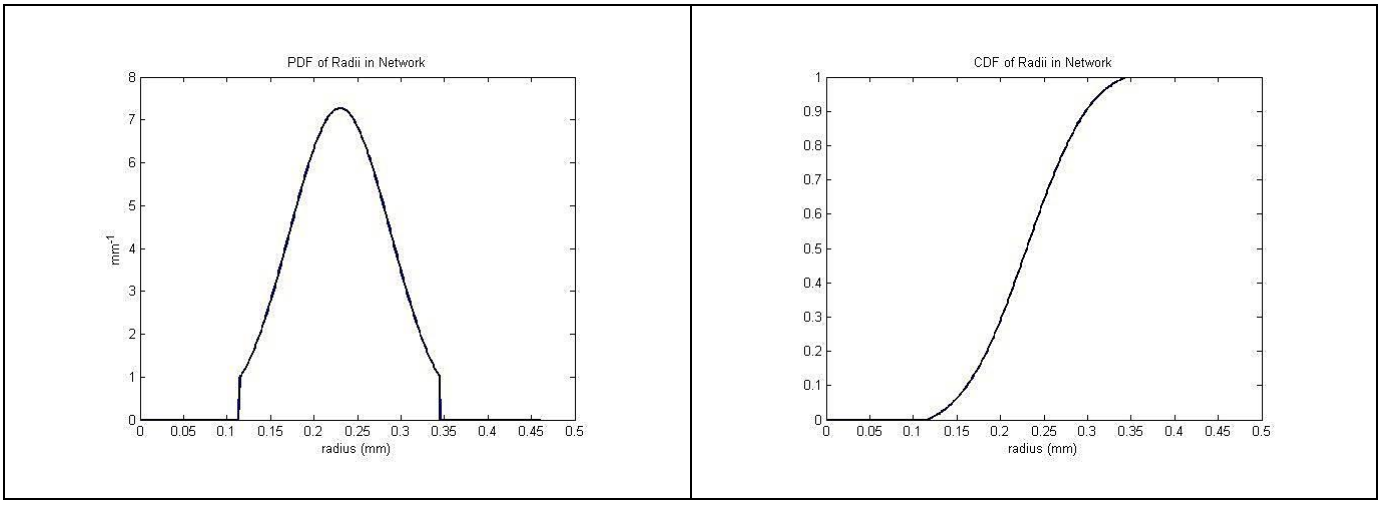


Figure 5.14: Truncated normal PDF and CDF for radii in network.

5.3.1 Stable Displacement Regime

The example stable displacement regime is given for $(\log M, \log C_a) = (4, 6)$. To attain these values, the simulations use $\mu_1 = 0.0001$, $\mu_2 = 1$, $\gamma = 0.03$, $\Sigma = 1.6264 \times 10^{-5}$ and $Q = 0.48792$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.15 while Figure 5.16 through Figure 5.19 shows the four additional plots of interest.

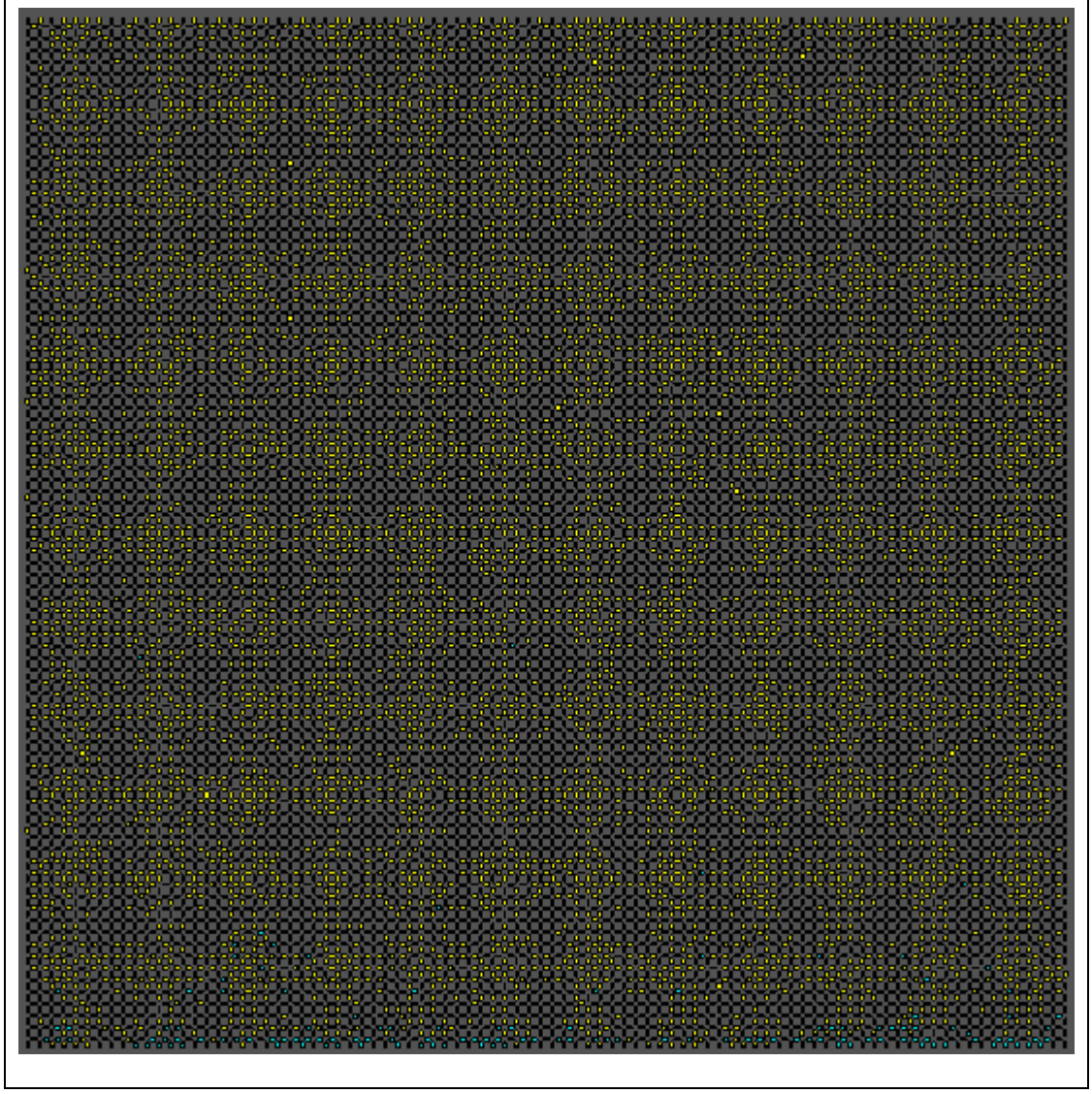


Figure 5.15: Stable displacement in truncated normal network.

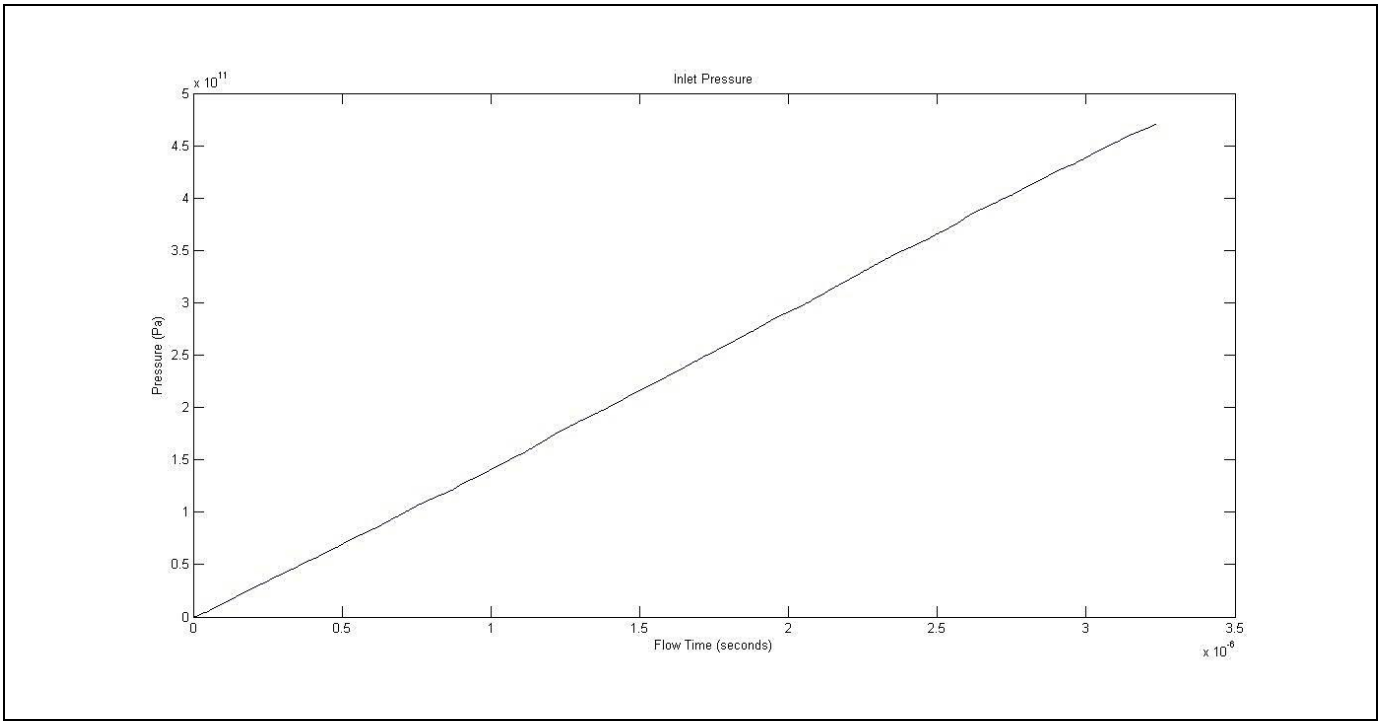


Figure 5.16: Inlet pressure for stable truncated normal network.

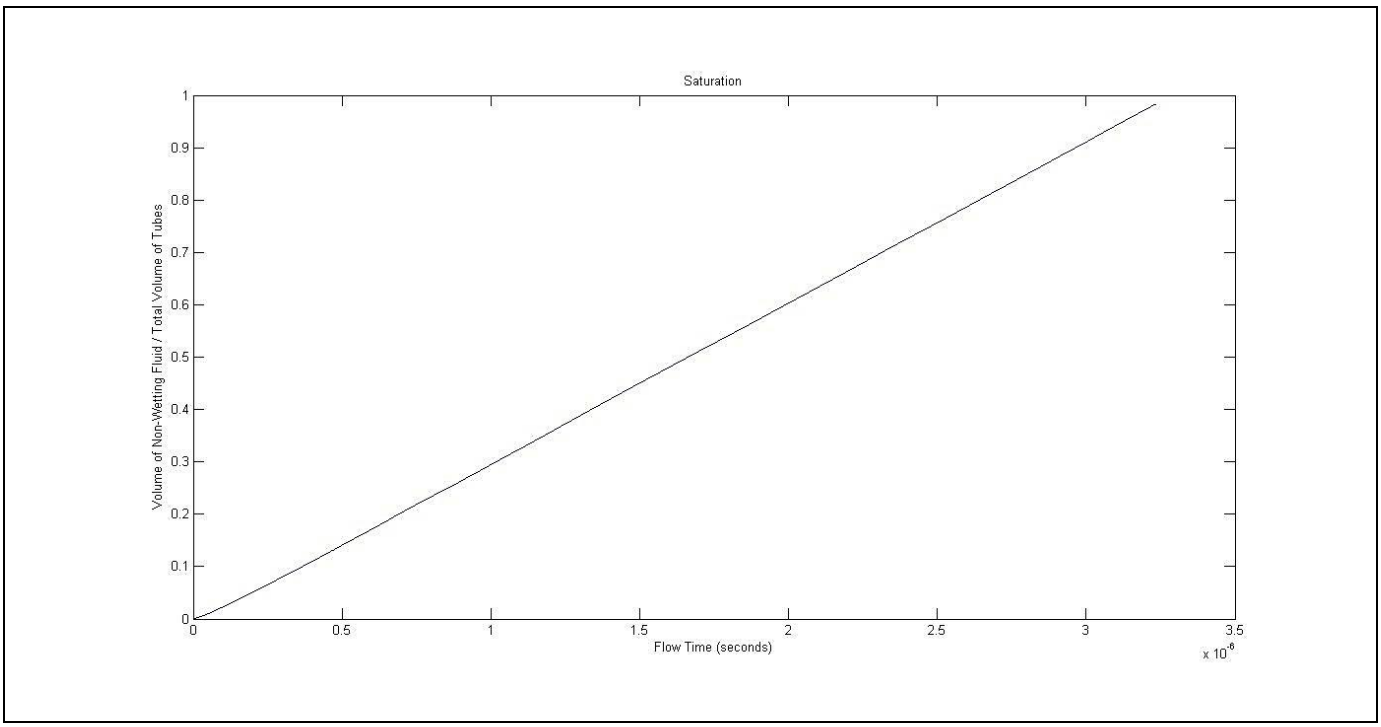


Figure 5.17: Saturation for stable truncated normal network.

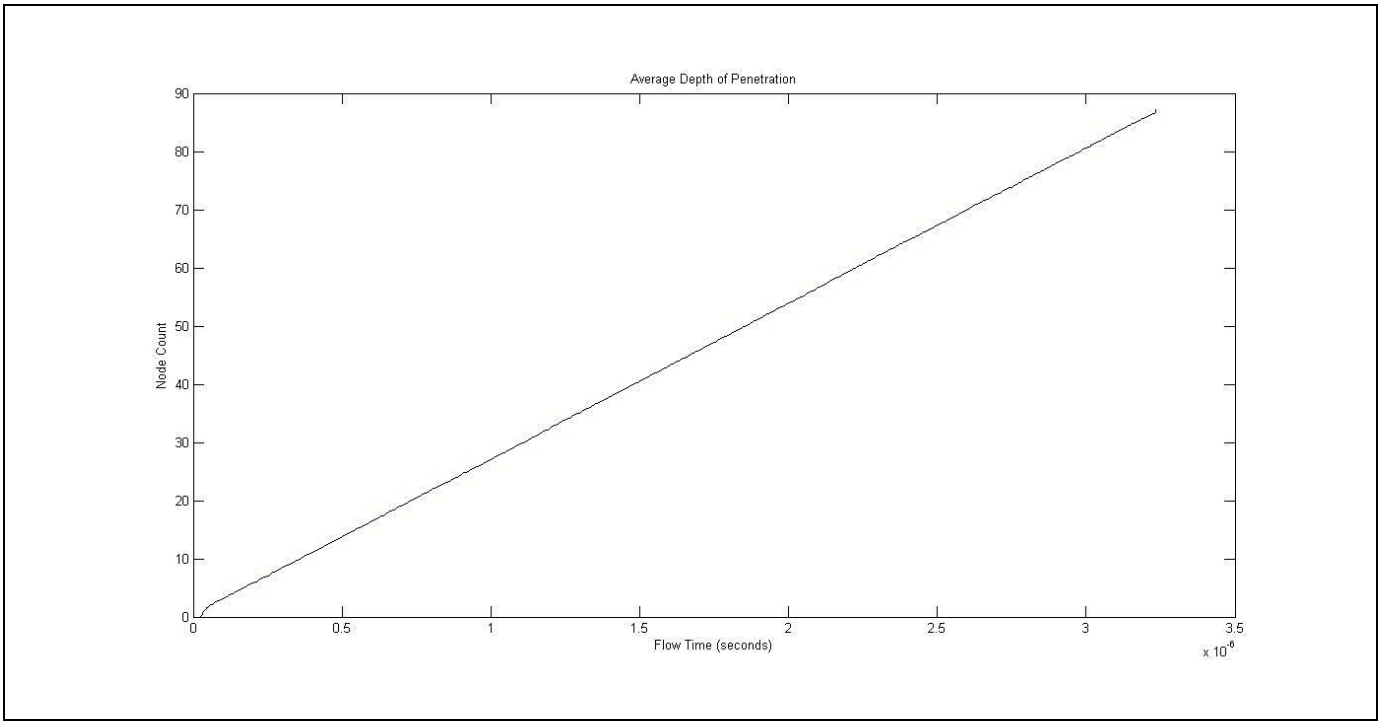


Figure 5.18: Average depth of penetration for stable truncated normal network.

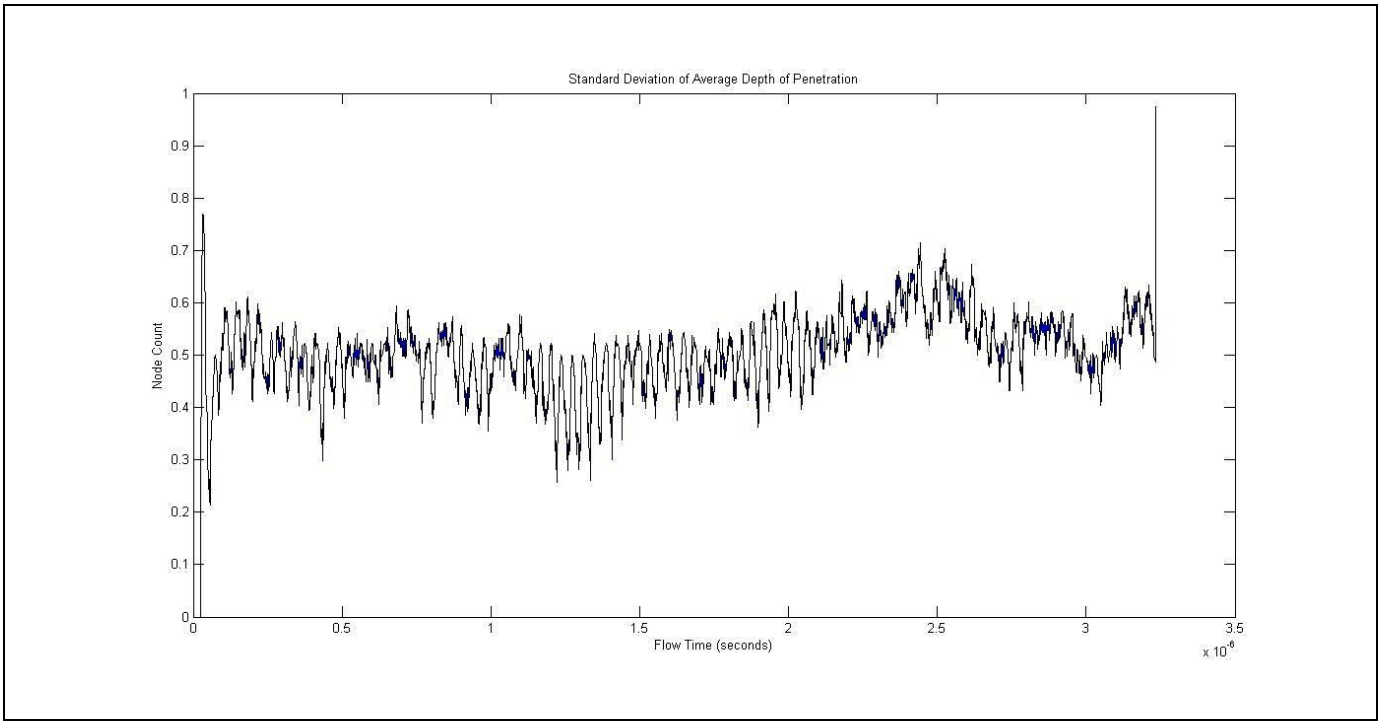


Figure 5.19: Standard deviation of average depth of penetration for stable truncated normal network.

5.3.2 Viscous Fingering Regimes

For the viscous fingering regime in the Lenormand diagram, three flow regime plots are shown. These plots, generated with a different seed for the truncated normal random number generator, show qualitatively the same information.

The first network that exhibits viscous fingering is given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations use $\mu_1 = 1$, $\mu_2 = 10^{-6}$, $\gamma = 0.03$, $\Sigma = 1.6264 \times 10^{-5}$ and $Q = 487920$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.20.

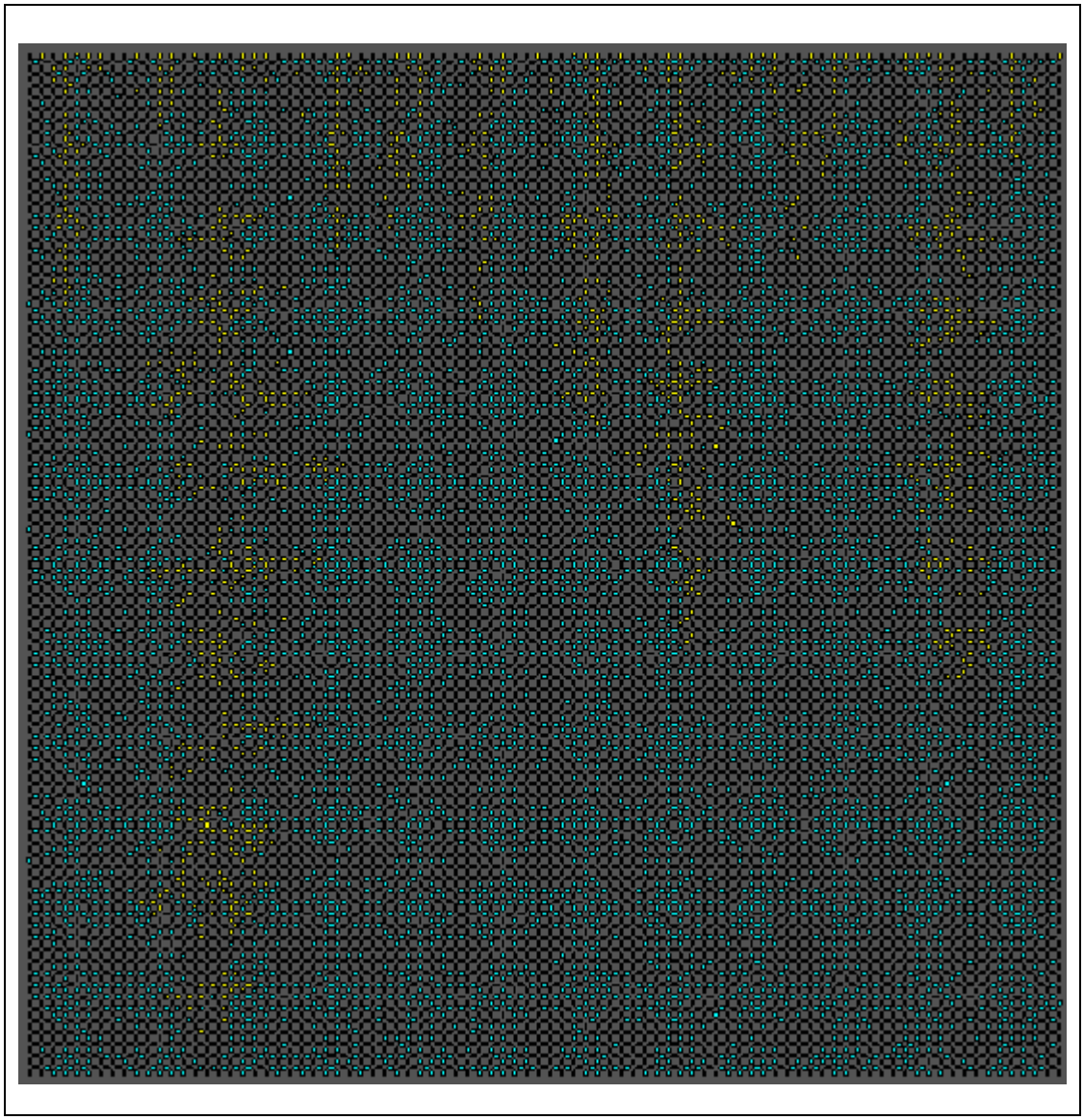


Figure 5.20: Viscous displacement in first truncated normal network.

The second network that exhibits viscous fingering is also given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations now use $\Sigma = 1.4090 \times 10^{-5}$ and $Q = 422700$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.21.

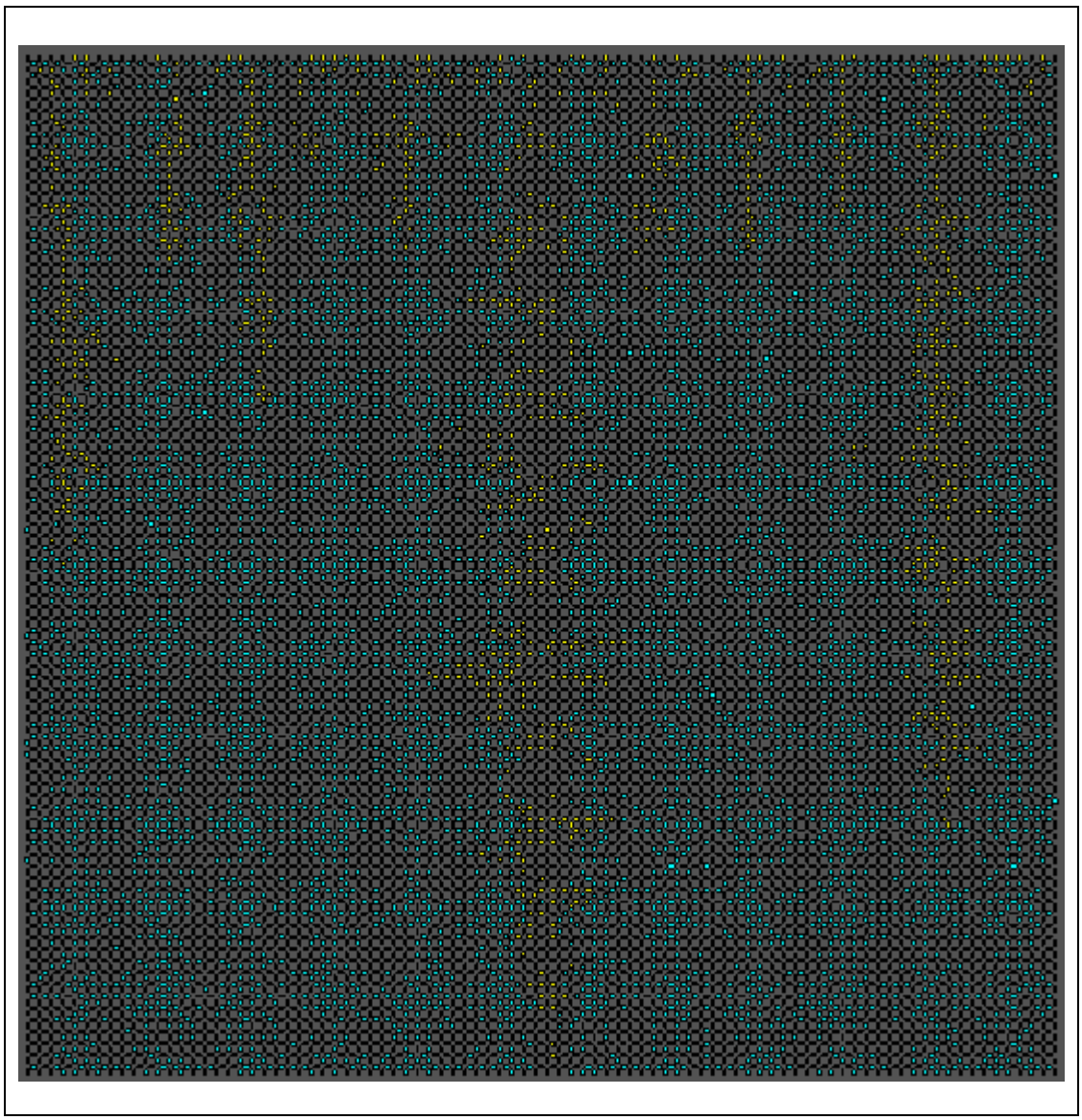


Figure 5.21: Viscous displacement in second truncated normal network.

The third network that exhibits viscous fingering is also given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations now use $\Sigma = 1.4898 \times 10^{-5}$ and $Q = 446940$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.22.

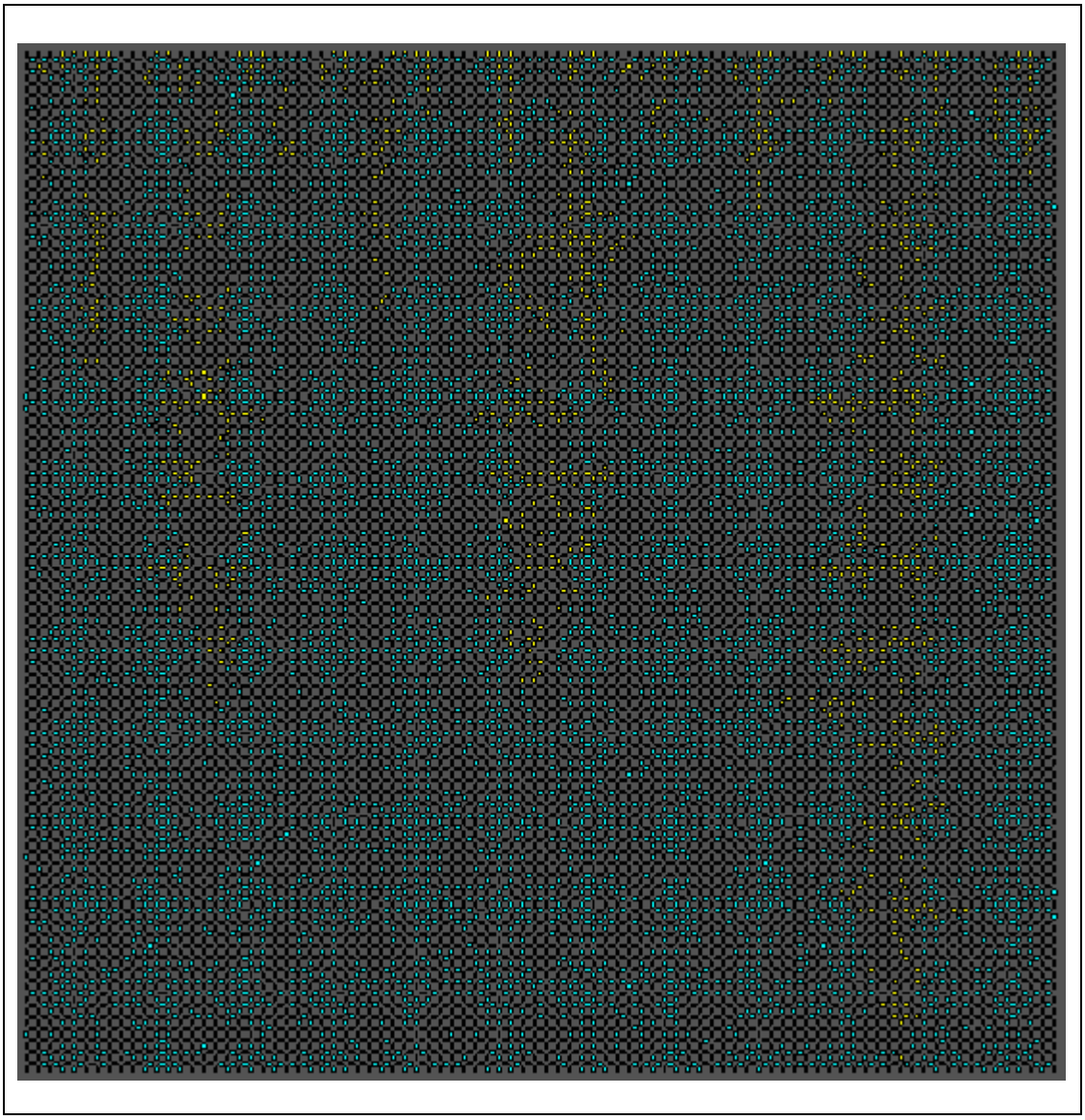


Figure 5.22: Viscous displacement in third truncated normal network.

For the sake of comparison, Figure 5.23 through Figure 5.26 shows the four additional plots for each viscous network on the same coordinate planes. As these plots show and the flow regimes support, there is qualitatively no significant difference caused by the seeding in the random number generator.

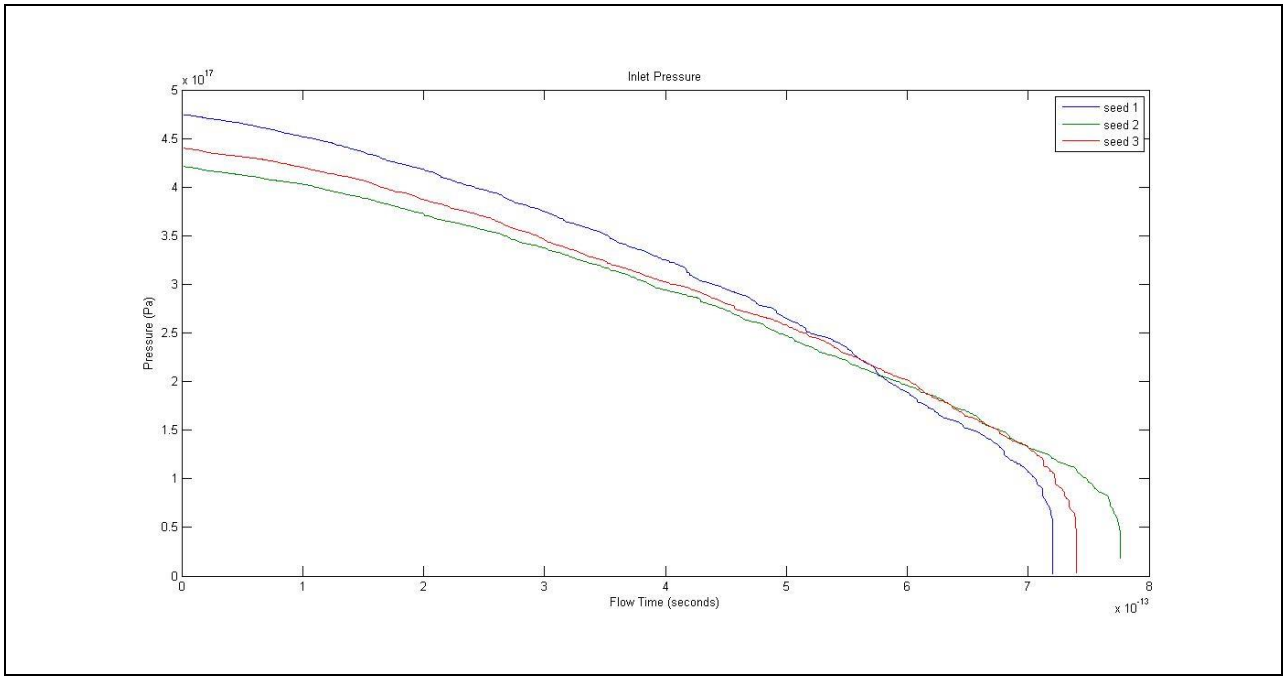


Figure 5.23: Inlet pressure for each viscous truncated normal network.

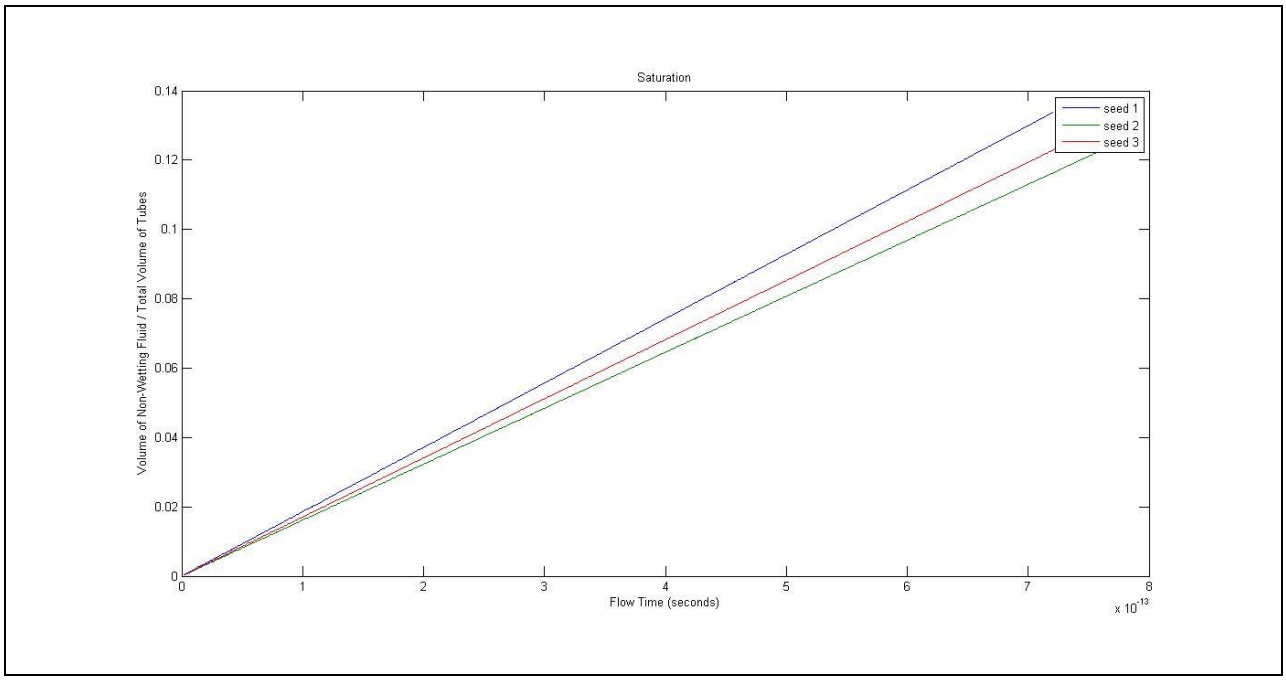


Figure 5.24: Saturation for each viscous truncated normal network.

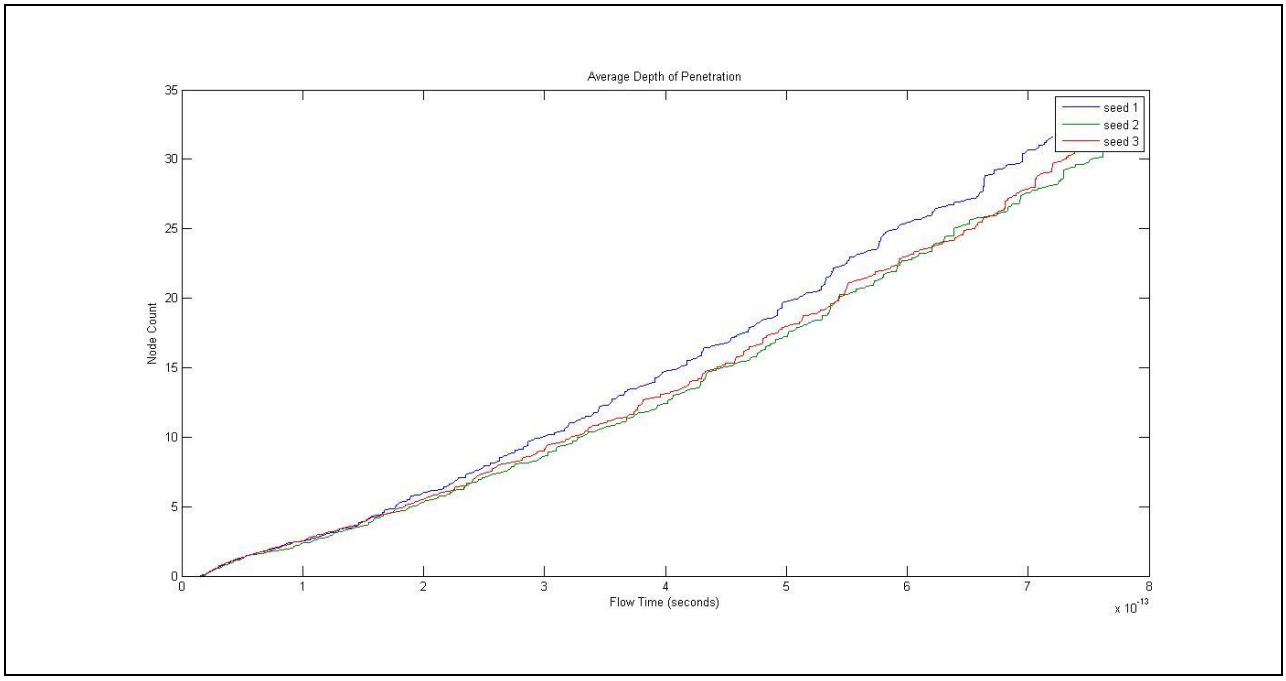


Figure 5.25: Average depth of penetration for each viscous truncated normal network.

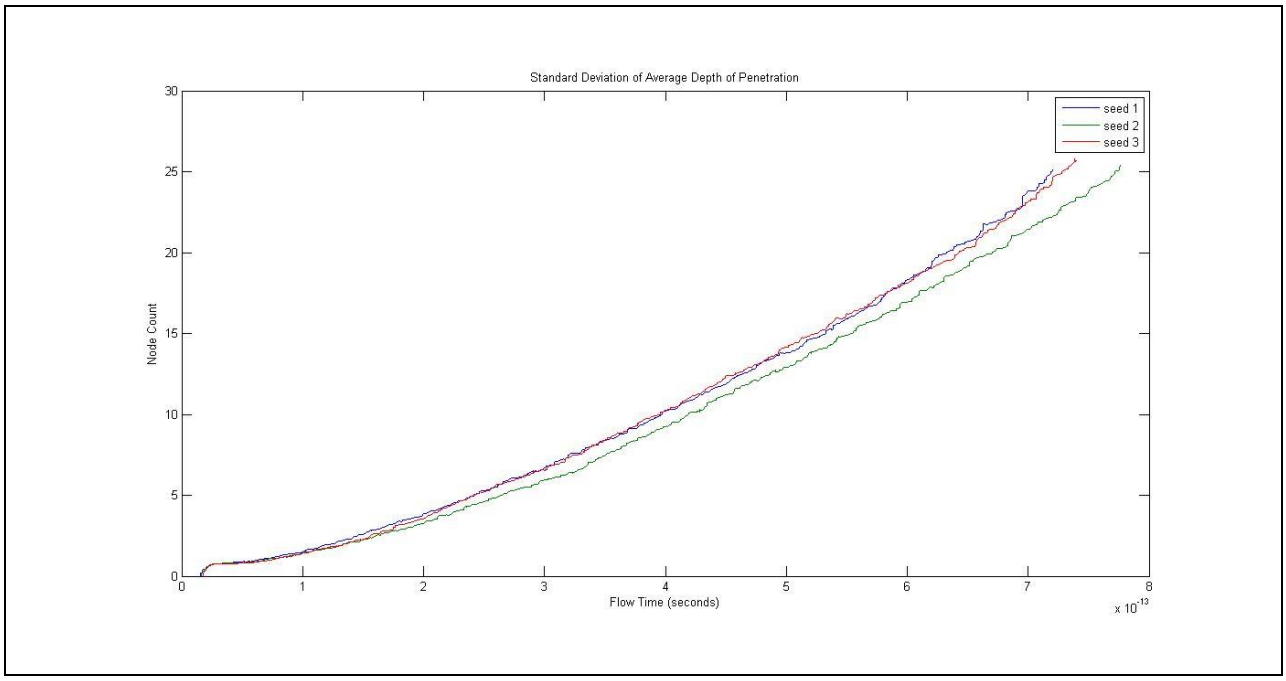


Figure 5.26: Standard deviation of average depth of penetration for each viscous truncated normal network.

5.4 Uniform Network

Figure 5.27 shows a uniform distribution for the radii of each network shown in this section.

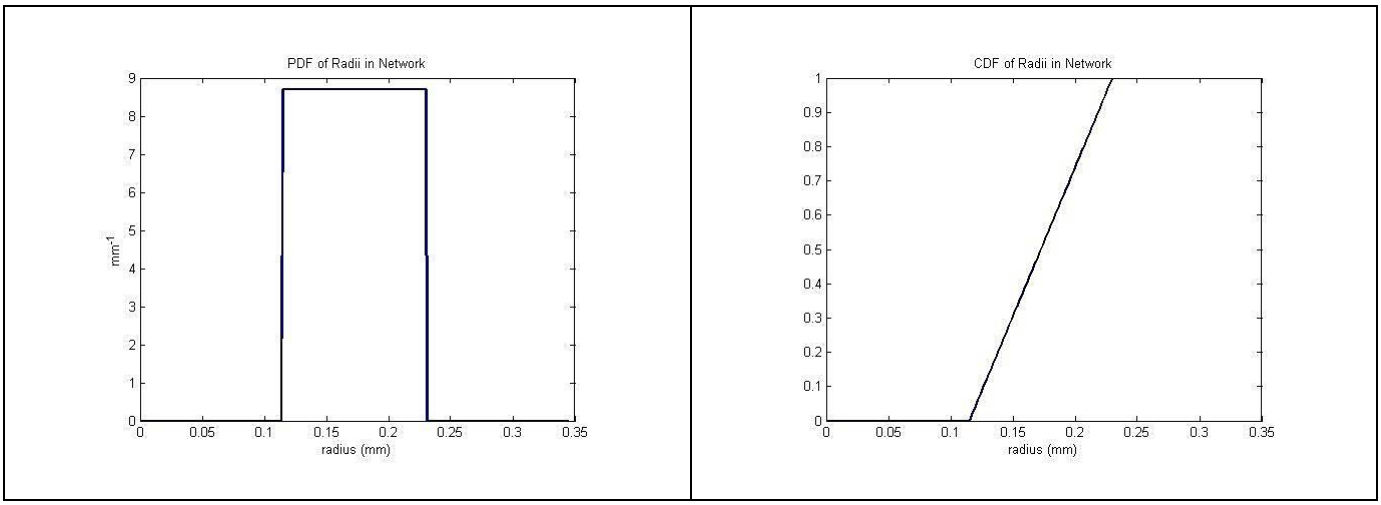


Figure 5.27: Uniform PDF and CDF for radii in network.

5.4.1 Stable Displacement Regime

The example stable displacement regime is given for $(\log M, \log C_a) = (4, 6)$. To attain these values, the simulations use $\mu_1 = 0.0001$, $\mu_2 = 1$, $\gamma = 0.03$, $\Sigma = 1.6730 \times 10^{-5}$ and $Q = 0.50190$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.28 while Figure 5.29 through Figure 5.32 shows the four additional plots of interest.

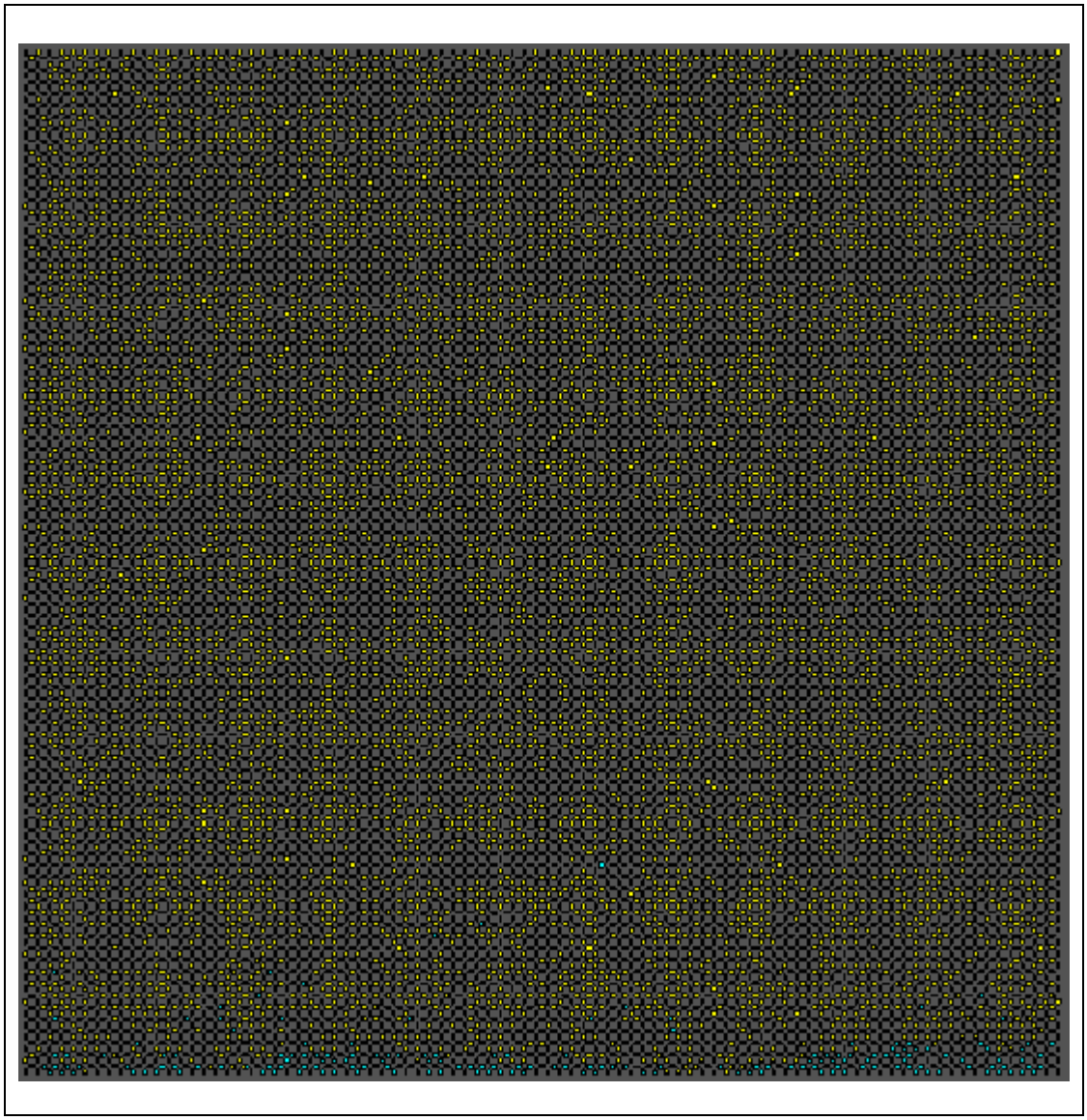


Figure 5.28: Stable displacement in uniform network.

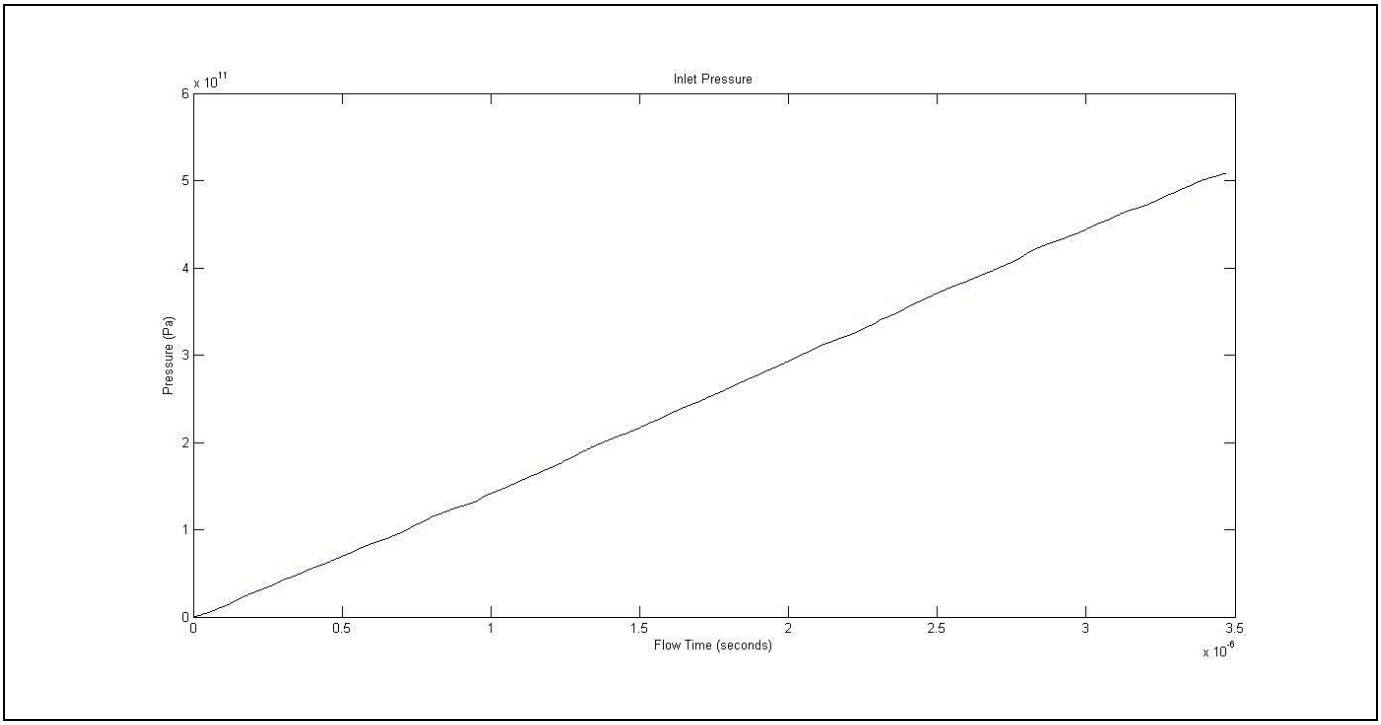


Figure 5.29: Inlet pressure for stable uniform network.

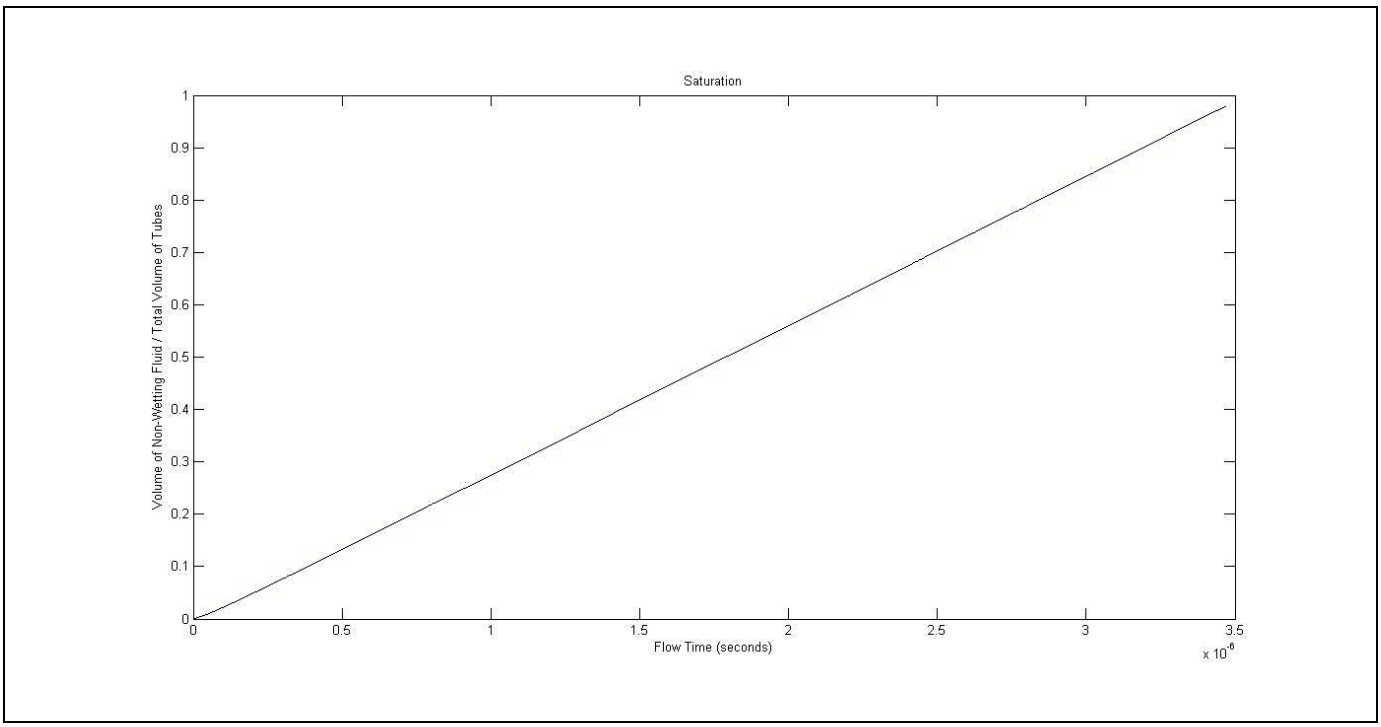


Figure 5.30: Saturation for stable uniform network.

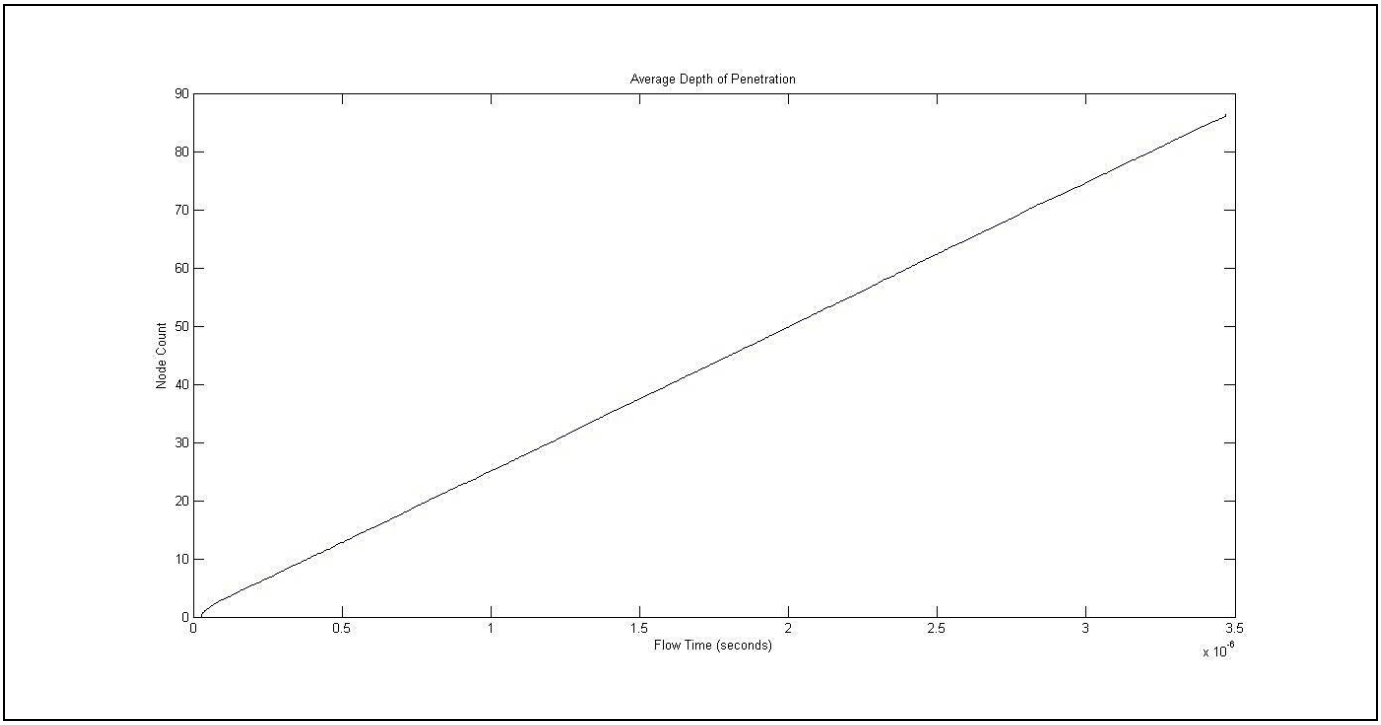


Figure 5.31: Average depth of penetration for stable uniform network.

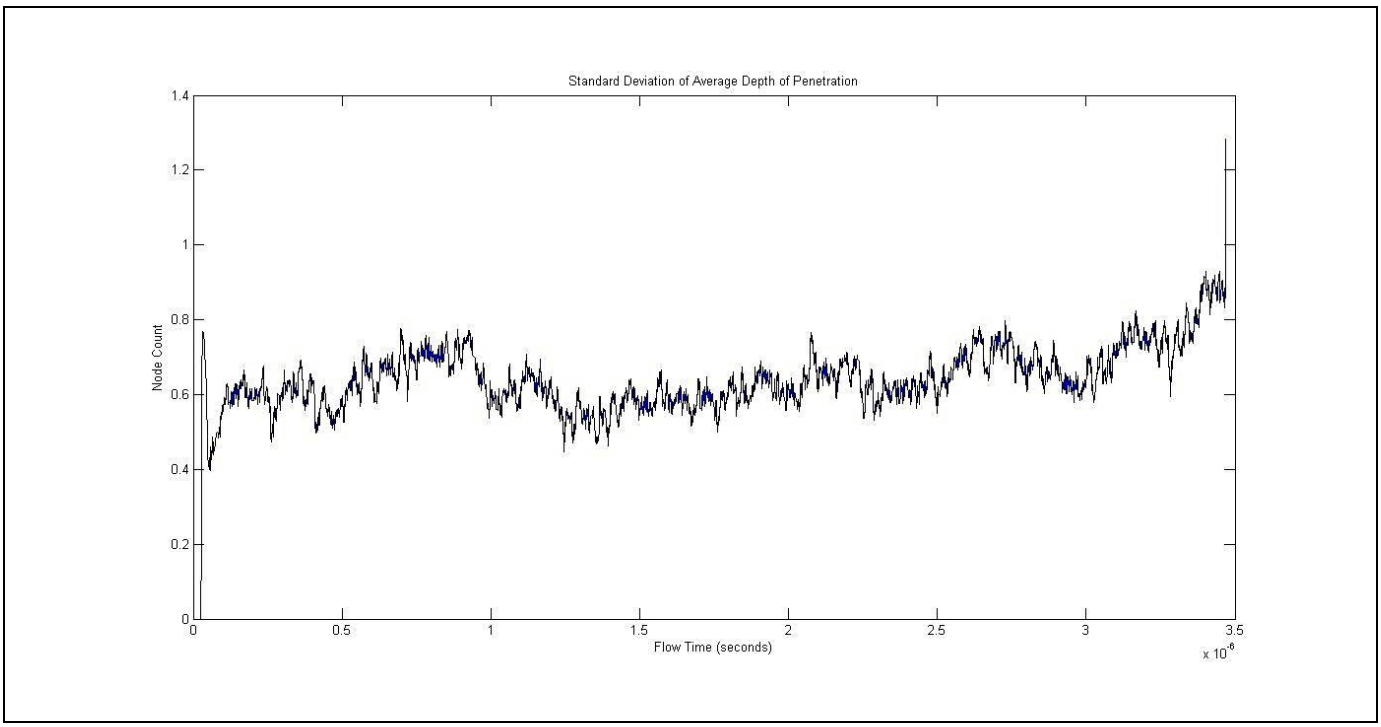


Figure 5.32: Standard deviation of average depth of penetration for stable uniform network.

5.4.2 Viscous Fingering Regimes

For the viscous fingering regime in the Lenormand diagram, three flow regime plots are shown. These plots, generated with a different seed for the uniform random number generator, show qualitatively the same information.

The first network that exhibits viscous fingering is given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations use $\mu_1 = 1$, $\mu_2 = 10^{-6}$, $\gamma = 0.03$, $\Sigma = 1.6730 \times 10^{-5}$ and $Q = 501900$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.33.

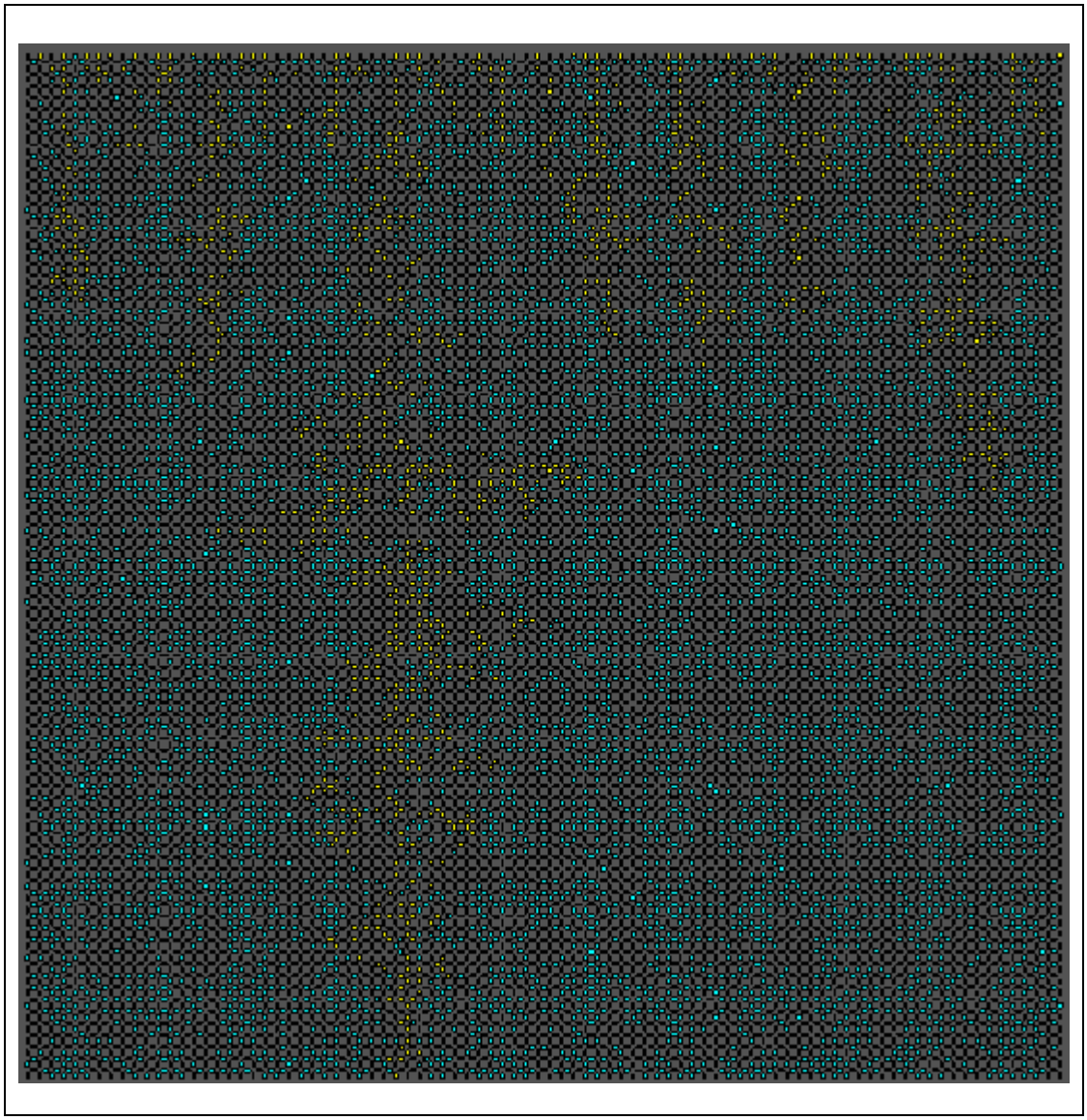


Figure 5.33: Viscous displacement in first uniform network.

The second network that exhibits viscous fingering is also given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations now use $\Sigma = 14190 \times 10^{-5}$ and $Q = 425700$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.34.

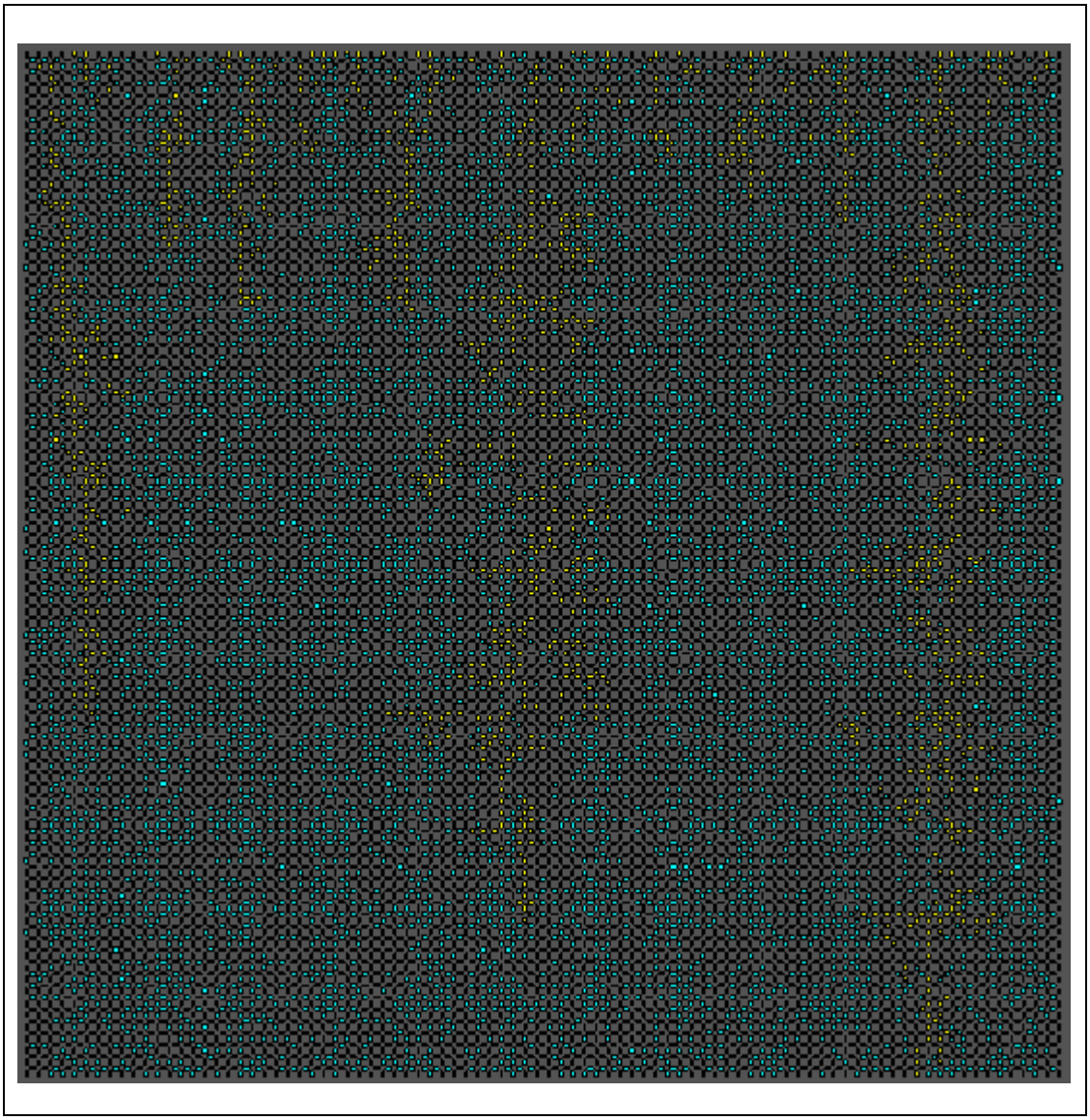


Figure 5.34: Viscous displacement in second uniform network.

The third network that exhibits viscous fingering is also given for $(\log M, \log C_a) = (-6, 6)$. To attain these values, the simulations now use $\Sigma = 1.5216 \times 10^{-5}$ and $Q = 456480$ in standard SI units. The snapshot of the network at breakthrough is shown in Figure 5.35.

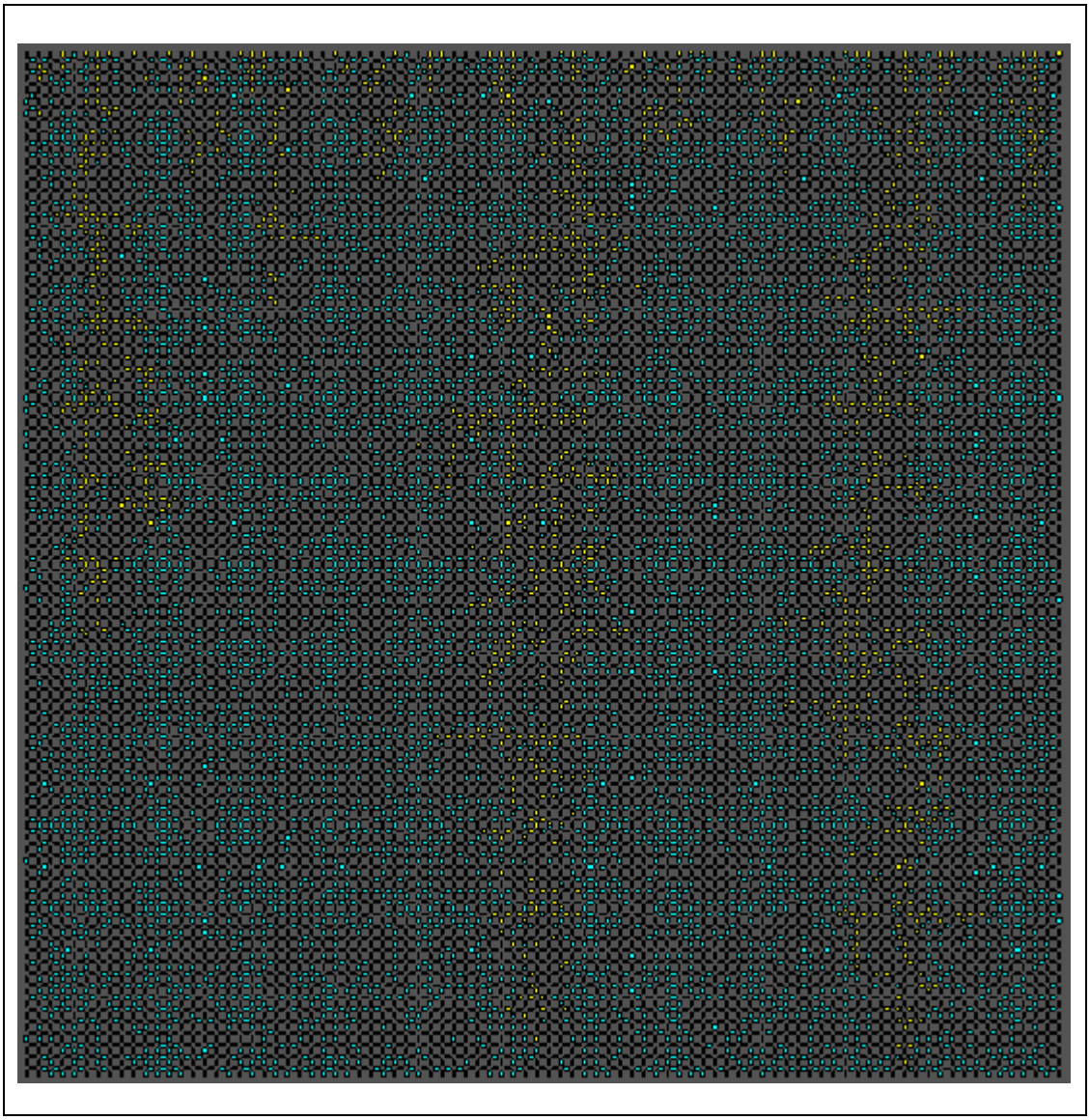


Figure 5.35: Viscous displacement in third uniform network.

For the sake of comparison, Figure 5.36 through Figure 5.39 shows the four additional plots for each viscous network on the same coordinate planes. As these plots show and the flow regimes support, there is qualitatively no significant difference caused by the seeding in the random number generator.

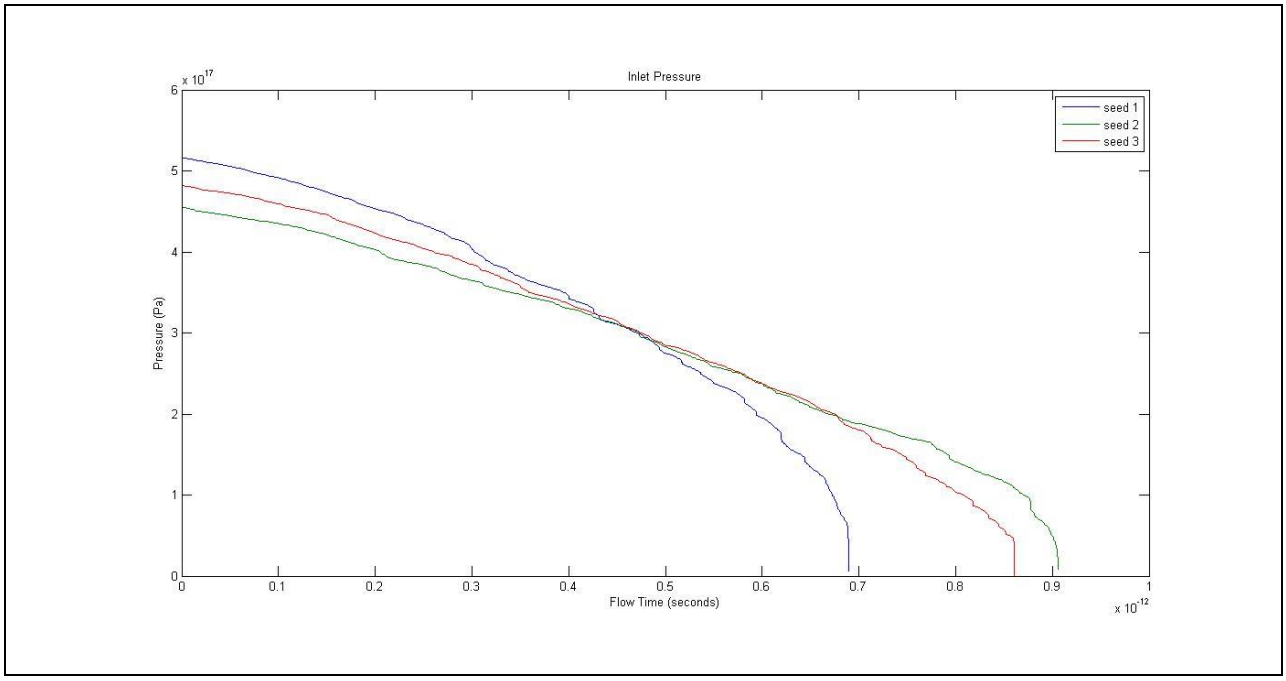


Figure 5.36: Inlet pressure for each viscous uniform network.

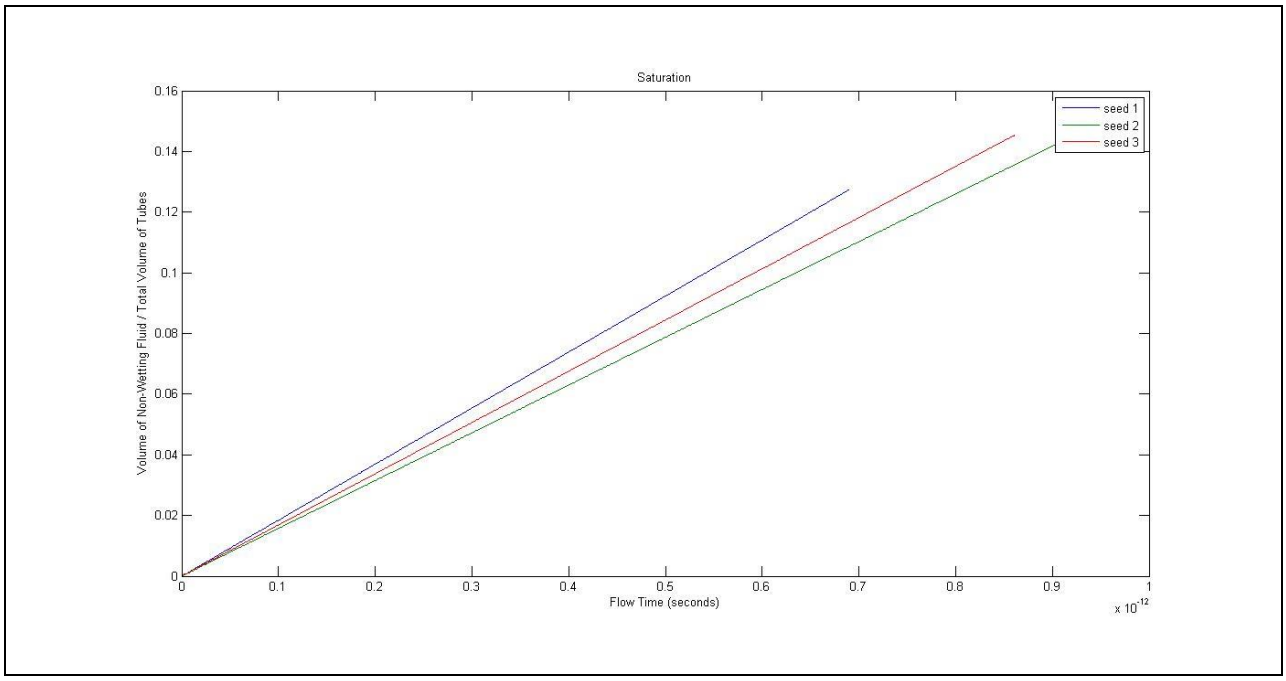


Figure 5.37: Saturation for each viscous uniform network.

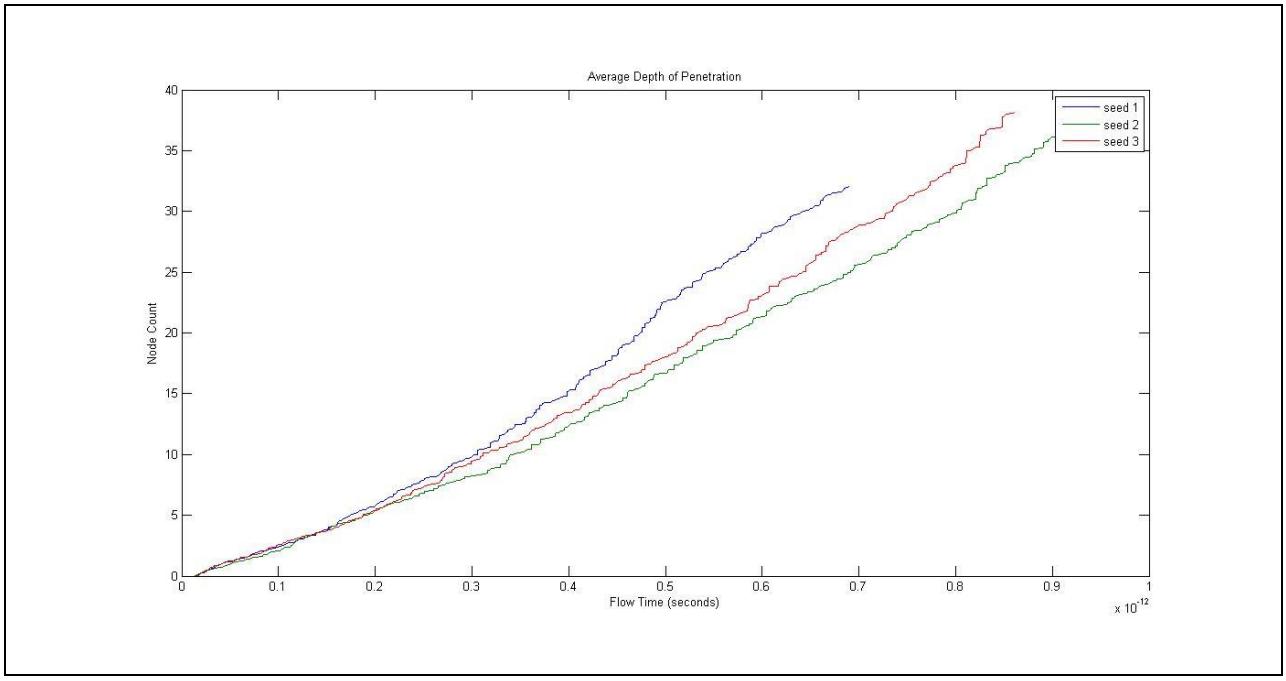


Figure 5.38: Average depth of penetration for each viscous uniform network.

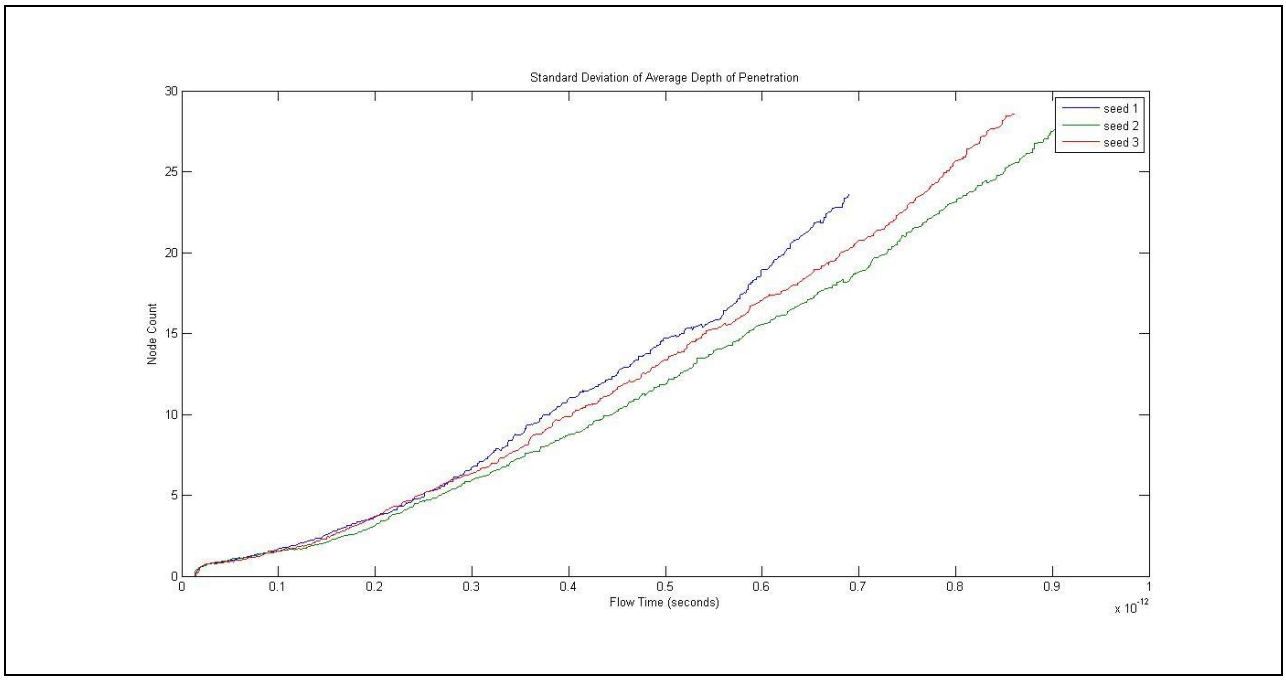


Figure 5.39: Standard deviation of average depth of penetration for each viscous uniform network.

5.5 Capillary Fingering Regime

For low capillary numbers, the model often terminates early with the infinite time step stopping condition. As a result of the apparent insufficient pressure gradient, the capillary fingering regime and part of the viscous fingering regime do not yield flow patterns.

At this point, the model is considered erroneous and alterations are necessary. Although the alterations discussed momentarily do not change the underlying issue, each one is worth mentioning. Because the nominal current of the source Q_N is not delivered, one attempted alteration involves solving the nodal pressures, summing the currents at the inlet (or outlet), and determining a correction current Q_C such that the desired current Q is met by adjusting the current delivered by the source to be $Q_N + Q_C$. However, since there is no guarantee this solves the issue, this updated procedure must be looped until the current is within some acceptable tolerance of the desired current. Using this approach, it is confirmed that MNA must be performed several times per iteration and is deemed highly inefficient.

The last alteration attempted essentially states that a tube is an evolvable circuit component. Essentially, all single-phase tubes are resistors while any two-phase tube is a resistor connected in series with a pressure source to model its capillary pressure as the nodal pressures are computed. However, for a general network, there are two possible configurations to represent each two-phase tube as it is unknown which terminal of the pressure source should be positive and which terminal should be negative. These possibilities are shown in Figure 5.40. Even ignoring this considerable drawback, for a trivial 3 by 2 network of five tubes, the first iteration produces negative nodal pressures at some nodes for all four possible networks (only two of the tubes at the beginning are multiphase). Although each of these possible networks delivered the desired injection rate, there remains the issue of picking one of the possible configurations and justifying the large number of times MNA must be computed for an iteration to locate the correct configuration.

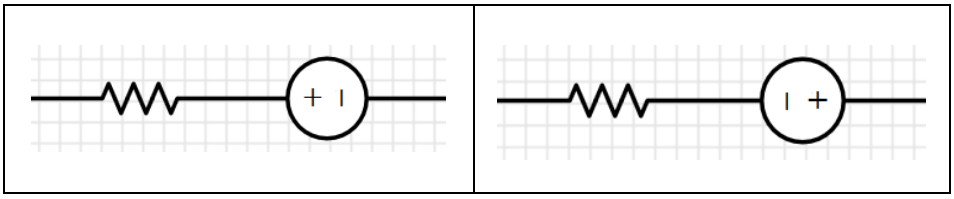


Figure 5.40: Two-phase tube possibilities.

Chapter 6: Closing Remarks

6.1 Conclusions

Attempting to match the flow regime patterns in the Lenormand diagram for low capillary numbers reveals that the specified injection rate is not met. In fact, an insufficient pressure gradient causes the algorithm to terminate prematurely while the system should produce an adequate pressure gradient to meet the nominal injection rate. To examine the source of error, recall that the algorithm assumes tubes are resistors when nodal pressures are computed. This assumption leads the algorithm to delay its treatment of capillary pressures until the moment flux, or current, is computed for each tube. However, as the results confirm, this decision leads to a violation of Kirchhoff's current law. As resistors, the sum of current entering each node equals the sum of current exiting the node. However, when capillary pressure is considered, some of the would-be positive currents become zero thereby breaking Kirchhoff's current law. Because segregating the treatments of nodal pressures and capillary pressures in the computation of current is flawed, capillary pressure cannot be handled with carefully outlined flow rules. Because it has a conductance and capillary pressure associated with it, each general tube is not a resistor and another circuit element is needed to consider the capillary pressures of tubes as nodal pressures are computed. Therefore, the merit of a circuit approach to two-phase flow in porous media relies on a better circuit equivalent for a tube. Additionally, while previous works arrive at flow patterns that qualitatively match the flow regimes in the Lenormand diagram, their approaches cannot be justified from a circuit perspective. For example, the simulator of Aker et al. essentially states that the desired injection rate is the sum of the injection given by Darcy's law for single-phase flow and the injection needed to overcome a so-called global capillary pressure. However, circuit theory does not support the notion of a global capillary pressure.

6.2 Future Work

Considering capillary pressure for the computation of nodal pressures during the modified nodal analysis stage of the algorithm needs to be addressed. There are a couple of potential approaches to handle this issue.

A switching element such as a diode would most likely model a two-phase tube more accurately. However, the diode is not one of the basic elements defined in the first stage of circuit analysis. Its circuit equivalent, derived from examining this semiconductor material at the atomic level, is a composite of other circuit components (Vladimirescu, 1994). At minimum, it is equivalent to a resistor connected in series with a current source that is connected in parallel with a capacitor. As such, each of these contributing components needs further examination to develop the analogy further. As can be concluded, any expansion of the basic set of allowable components requires modification to include those elements, which is particularly relevant during the modified nodal analysis stage of the algorithm.

Because it is not guaranteed that a tube can be related to a diode, the attempted alteration which essentially states that a tube is a resistor connected in series with a voltage source to model its capillary pressure might be the most approachable. However, there are a few drawbacks associated with this attempt. As stated in the previous chapter, there is an inherent issue in picking the polarity for these voltage sources. In order to make this approach workable, the algorithm would have to dictate that the voltage source be situated in the same manner each time one is introduced into a tube. While sensible, this makes backflow difficult if not impossible to handle. Another drawback is that the network would continually evolve. Although a tube would always maintain a resistor, the network would have to be able to insert and remove one or more voltage sources; when a multiphase tube fills to become single phase, it would lose its voltage source and one or more other tubes would have to introduce voltage sources into their circuit equivalents to handle their capillary pressures. This dynamic capability would only be possible if a nodal numbering technique could be developed and reliably applied such that the network did not change too much between iterations. Although algorithmically challenging, this approach would be advantageous in that no additional elements would have to be added to the basic set of allowable components because voltage sources, resistors, and current sources are already defined.

Bibliography

Abdallah, W., Buckley, J.S., Carnegie, A., Edwards, J., Herold, B., Fordham, E., Graue, A., Habashy, T., Seleznev, N., Signer, C., Hussain, H., Montaron, B., and Ziauddin, M., 1986, "Fundamentals of Wettability," Technology, **38**, pp. 1125-1144.

Aker, E., Måløy, K.J., Hansen, A., and Batrouni, G.G., 1998, "A Two-Dimensional Network Simulator for Two-Phase Flow in Porous Media," Transport in Porous Media, **32**(2), pp. 163-185.

Al-Gharbi, M.S.and Blunt, M.J., 2005, "Dynamic network modeling of two-phase drainage in porous media," Physical Review E, **71**(1), pp. 016308(1)-016308(16).

Bear, J., 1972. *Dynamics of Fluids in Porous Media*, American Elsevier Publishing Company Inc., New York, Chap. 1.

Bear, J. and Bachmat, Y., 1991. *Introduction to Modeling of Transport Phenomena in Porous Media*, Kluwer Academic Publishers, Boston, Chap. 1.

Chapra, S.C. and Canale, R.P., 2010. *Numerical Methods for Engineers*, 6th ed., McGraw-Hill, New York, NY, Chap. 23.

Cheever, E., n.d., "An Algorithm for Modified Nodal Analysis," from http://www.swarthmore.edu/NatSci/echeeve1/Ref/mna/MNA3.html

Chen, J. and Wilkinson, D., 1985, "Pore-Scale Viscous Fingering in Porous Media," Physical Review Letters, 55(18), pp. 1892-1896.

Chua, L.O. and Lin P., 1975. *Computer-Aided Analysis of Electronic Circuits: Algorithms & Computational Techniques*, Prentice-Hall Inc., Englewood Cliffs, NJ, Chap. 1 – 2.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., 2009. *Introduction to Algorithms*, 3rd ed., The MIT Press, Cambridge, Massachusetts, Chap. 22.

Fatt, I., 1956, "The network model of porous media," AIME Petroleum Transactions, pp. 144-159.

Ferer, M., Geisbrecht, R.A., Sams, W.N., and Smith, D.H., 1992, "Crossover from fractal to compact growth from simulations of two-phase flow with finite viscosity ratio in two-dimensional porous media," Physical Review A, **45**(10), pp. R6973.

Ferer, M., Sams, W.N., Geisbrecht, R.A., and Smith, D.H., 1993, "Crossover from fractal to compact flow from simulations of two-phase flow with finite viscosity ratio in two-dimensional porous media," Physical Review E, **47**(4), pp. 2713.

Ferer, M., Bromhal, G.S., and Smith, D.H., 2002, "Pore-level modeling of immiscible drainage: validation in the invasion percolation and DLA limits," Physica A: Statistical Mechanics and its Applications, **319**, pp. 11-35.

Ferer, M., Shelley, L.A., Tortora, P., Kadambi, J.R., Oliver, M., Bromhal, G.S., and Smith, D.H., 2011, "Two-Phase Flow in Porous Media: Predicting Its Dependence on Capillary Number and Viscosity Ratio," Transport in Porous Media, **86**(1), pp. 243-259.

Heineman, G.T., Pollice, G., and Selkow, S., 2008. *Algorithms in a Nutshell*, 1st ed., O'Reilly Media, Sebastopol, CA, Chap. 8.

Ho, C., Ruehli, A.E., and Brennan, P.A., 1975, "The Modified Nodal Approach to Network Analysis," IEEE Trans. Circuits Syst. **CAS-22**, pp. 504-509.

Hughes, J.F., Van Dam, A., McGuire, M., Sklar, D.F., Foley, J.D., Feiner, S.K., and Akeley, K., 2013. *Computer Graphics: Principles and Practice*, Addison-Wesley Professional, Upper Saddle River, NJ, Chap. 7-10.

Ioinovici, A., 1990. *Computer-aided Analysis of Active Circuits*, Marcel Dekker Inc., New York, New York, Chap. 2 – 3.

Joekar-Niasar, V. and Hassanizadeh, S.M., 2012, "Analysis of two-phase flow in porous media using dynamic pore-network models: A review," Critical Reviews in Environmental Science and Technology, 42(18), pp. 1895-1976.

Kincaid, D. and Cheney, W., 2002. *Numerical Analysis: Mathematics of Scientific Computing*, 3rd ed., American Mathematical Society, Pacific Grove, CA, Chap. 8 – 9.

Knudsen, H.A., Aker, E., and Hansen, A., 2001, "Bulk Flow Regimes and Fractional Flow in 2D Porous Media by Numerical Simulations," Transport in Porous Media, **47**(1), pp. 99-121.

Lenormand, R., Zarcone C., and Sarr, A., 1983, "Mechanisms of the Displacement of One Fluid by Another in a Network of Capillary Ducts," Journal of Fluid Mechanics, **135**, pp. 337-353.

Lenormand, R. and Zarcone, C., 1985, "Invasion Percolation in an Etched Network: Measurement of a Fractal Dimension," Physical Review Letters, **54**, pp. 2226-2229.

Lenormand, R., 1987, "Statistical Physics and Immiscible Displacements through Porous Media," Physics and Chemistry of Porous Media II, **154**(1), pp. 98-115.

Lenormand, R., Touboul, E., and Zarcone, C., 1988, "Numerical Models and Experiments on Immiscible Displacements in Porous Media," Journal of Fluid Mechanics, **189**, pp. 165-187.

Lenormand, R. and Zarcone, C., 1989, "Capillary Fingering: Percolation and Fractal Dimension," Transport in porous media, **4**(6), pp. 599-612.

Lenormand, R., 1990, "Liquids in Porous Media," Journal of Physics: Condensed Matter, **2**(S), pp. SA79-SA88.

Måløy, K.J., Feder J., and Jøssang, T., 1985, "Viscous Fingering Fractals in Porous Media," Physical Review Letters, **55**(24), pp. 2688-2691.

Mitcheson, P.D., "EE 2.3: Semiconductor Modelling in SPICE," from http://www3.imperial.ac.uk/pls/portallive/docs/1/7292571.PDF

Nilsson, J. W. and Riedel, S. A., 2008. *Electric Circuits*, 8th ed., Pearson Prentice Hall, Upper Saddle River, NJ, Chap. 2 – 3.

Office of Fossil Energy, "Enhanced Oil Recovery," from http://energy.gov/fe/science-innovation/oil-gas/enhanced-oil-recovery

Parent, R., 2012. *Computer Animation*, Morgan Kaufmann, Chap. 2.

Pederson, D.O., 1984, "A Historical Review of Circuit Simulation," IEEE Trans. Circuits Syst., **CAS-31**, pp. 103-111. Petersen, R.T., 2009, "Pore-Scale Modeling of the Impact of Surrounding Flow Behavior on Multiphase Flow Properties," M.S. thesis, University of Texas at Austin.

Petersen, R.T., 2009, "Pore-Scale Modeling of the Impact of Surrounding Flow Behavior on Multiphase Flow Properties," M.S. thesis, University of Texas at Austin.

Phillips, O.M., 1991. *Flow and Reactions in Permeable Rocks*, Cambridge University Press, New York, Chap. 2. Reeves, P.C. and Celia, M.A., 1996, "A Functional Relationship between Capillary

Pressure, Saturation, and Interfacial Area as Revealed by a Pore-Scale Network Model," Water Resources Research, **32**(8), pp. 2345-2358.

Reeves, P.C. and Celia, M.A., 1996, "A Functional Relationship between Capillary Pressure, Saturation, and Interfacial Area as Revealed by a Pore-Scale Network Model," Water Resources Research, **32**(8), pp. 2345-2358.

Reza, M. and Pillai, K.M., 2013, *Wicking in Porous Materials: Traditional and Modern Modeling Approaches*, Taylor & Francis Group, Boca Raton, FL, Chap. 1.

Sahimi, M., 1995. *Flow and Transport in Porous Media and Fractured Rock*, VHC Verlagsgesellschaft mbH, New York, Chap. 1.

Sedra, A.S. and Smith, K.C., 2004. *Microelectronic Circuits*, Oxford University Press, New York, New York, Chap. 3.

Sewell, G., 2005. *The Numerical Solution of Ordinary and Partial Differential Equations*, 2nd ed., John Wiley & Sons Inc., Hoboken, NJ, Chap. 1.

Stevenson, K., Ferer, M., Bromhal, G.S., Gump, J., Wilder, J., and Smith, D.H., 2006, "2-D network model simulations of miscible two-phase flow displacements in porous media: Effects of heterogeneity and viscosity," Physica A: Statistical Mechanics and its Applications, **367**, pp. 7-24.

Streetman, B.G. and Banerjee, S.K., 2006. *Solid State Electronic Devices*, Pearson Prentice Hall, Upper Saddle River, NJ, Chap. 1 – 5.

Vladimirescu, A., 1994. *The SPICE Book*, John Wiley & Sons Inc., New York, Chap. 3.

Warwick, C., 2009, "In a Nutshell: How SPICE Works," from http://www.emcs.org/acstrial/newsletters/summer09/HowSpiceWorks.pdf

Young, H.D., Freedman, R.A. and Ford, A.L., 2004. *Sears and Zemansky's University Physics: with Modern Physics*, 11th ed., Pearson Education Inc., San Francisco, CA, Chap. 25 – 26.

Curriculum Vita

Todd Wayne Dorethy Jr. was born in El Paso, Texas in 1988. As a Pre-Engineering Magnet student at Captain John L. Chapin High School, he graduated in the top ten percent in 2006. He was awarded the Presidential Excellence Scholarship from the University of Texas at El Paso and entered in the fall. He earned multiple appearances on the Dean's List and in 2009 he was awarded the Columbia Crew Memorial Scholarship. He graduated *cum laude* with a bachelor's degree in Electrical Engineering in 2010. He began his master's degree in Computational Science in 2012 and graduated in 2014. As a member of the university's centennial graduating class, he received the Computational Science Departmental Award for Academic and Research Excellence. While pursuing his master's degree, he worked as a research assistant in the Mechanical Engineering department at UTEP under the guidance of Dr. Vinod Kumar. In summer 2012, he worked alongside Dr. Kumar and research scientist Dr. Rao Gudimetla under a fellowship with the Air Force Research Laboratory in Kihei, Hawaii to model the effects of atmospheric turbulence on laser beam propagation. In summer 2013, as a Directed Energy scholar, he returned to Maui to continue the turbulence models.

Permanent Address: twdorethy@miners.utep.edu