Universitetet i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Petroleum Technology / Drilling Technology	Spring semester, 2017 Open
Author: Kristian Lie Vorre	…… (signature of author)
Programme coordinator: Jann Rune Ursin Supervisor(s): Yen Adams Sokama-Neuyam	
Title of master's thesis: Theoretical modelling of the effect of salt precipitation on CO2 injectivity	
Credits: 30	
Keywords: Permeability Salt Precipitation CO_2 Injectivity CCS Climate	Number of pages: 48 + supplemental material/other: 8 Stavanger, 15.06.2017 date/year

Abstract

Since the industrial revolution we have produced more carbon dioxide. Carbon dioxide is greenhouse gas which contributes to global warming. Carbon Capture & Storage (CCS) has been working on technical solutions that will capture carbon dioxide, which is being produced as a product of industrialization. This captured carbon dioxide will be transported to suited storage areas. These storage areas can be geological formation, often old oil & gas reservoirs. For the CCS to safely store the carbon dioxide, it needs to be transported to the location, and then injected into the reservoir. Injection requires knowledge, as the deposition sites can be more than a kilometer below the ground.

This thesis will focus on creating a model to simulate CO_2-injection. The model is created in MATLAB, and will simulate problems such as: colloidal fine particles and salt precipitation. This model will help to understand how a rock formation act during injection. Absolute permeabilities and relative imjectivity are the key results that will be measured by this model.

Acknowledgement

This thesis is written at the Department of Petroleum Engineering at the University of Stavanger.

Firstly, I would like to express my gratitude towards my external supervisor, Yen Adams Sokama-Neuyam, for answering the questions I had regarding this topic and providing me with literature as well as his guidance.

I would also like to thank my fellow students at the University of Stavanger for a great experience during my studies.

Table of Content

ABSTRACT ..... **III**

ACKNOWLEDGEMENT ..... **V**

TABLE OF CONTENT ..... **VII**

LIST OF TABLES ..... **IX**

LIST OF FIGURES ..... **XI**

NOMENCLATURE ..... **XIII**

ABBREVIATIONS XIII
SYMBOLS XIV

1 INTRODUCTION **1**

2 PROBLEM DEFINITION AND OBJECTIVES **2**

3 THEORY **3**

3.1 CLIMATE CHANGE AND CCS 3
3.2 GENERAL CCS CHALLENGES 4
3.3 MECHANISMS OF CO_2 INJECTIVITY IMPAIRMENT 5
3.3.1 Salt precipitation during CO_2 injection *5*
3.3.2 Fines Migration *8*
3.4 PORE-SCALE MODELLING 11
3.5 FLOW PATTERNS 13
3.5.1 Laminar *13*
3.5.2 Turbulent *13*
3.5.3 Brownian movements *13*

4 MODEL DEVELOPMENT **14**

4.1 FUNDAMENTAL ASSUMPTIONS AND SIMPLIFICATIONS 14
4.1.1 Overview *14*
4.1.2 Fluid flow through a single tube *15*
4.1.3 Fluid flow through N amount of tubes *16*
4.1.4 Injectivety impairment induced by salt precipitation *16*
4.1.5 Total number of capillary tubes, N *17*
4.1.6 Thickness of precipitated salt, Δr *18*
4.1.7 Absolute permeability *20*
4.2 OVERVIEW OF EQUATIONS USED IN THE MODEL 22
4.3 STATISTICAL MODELLING OF PARTICLE TRANSPORT 24
4.3.1 Laminar flow estimation *24*
4.3.2 Turbulent flow estimation *25*
4.4 COMPUTATIONAL ALGORITHM 26
4.4.1 MATLAB algorithm "TurbulentNPart.m" *26*
4.4.2 MATLAB algorithm "LaminarNPart.m" *28*
4.5 INPUTS FOR THE MATLAB-SCRIPTS 29

5 RESULTS AND DISCUSSION **30**

5.1 OVERVIEW OF RESULTS PRESENTATION 30
5.1.1 Turbulent flow - Porosity 0.100 ..... *30*
5.1.2 Turbulent flow - Porosity 0.184 ..... *32*
5.1.3 Turbulent flow - Porosity 0.300 ..... *33*
5.1.4 Laminar flow - Porosity 0.100 ..... *35*
5.1.5 Laminar flow - Porosity 0.184 ..... *37*
5.1.6 Laminar flow - Porosity 0.300 ..... *38*
5.2 PORE-SIZE DISTRIBUTION 39
5.3 EFFECT OF PARTICLE SIZE 40
5.4 EFFECT OF INITIAL PERMEABILITY 42

6 CONCLUSION 44

6.1 SUMMARY AND HIGHLIGHTS 44
6.2 PROPOSED FURTHER WORK 45

7 REFERENCES 46

8 APPENDIX 49

8.1 APPENDIX A: MATLAB-SCRIPT "TURBULENTNPART.M" 49
8.2 APPENDIX B: MATLAB-SCRIPT "LAMINARNPART.M" 52

List of Tables

Table 4-1: Input parameters for the core and tube in the MATLAB script 29
Table 4-2: Input parameters for the particles in the MATLAB script 29
Table 4-3: Input parameters for a low salt concentration..... 29
Table 4-4: Input parameters for a high salt concentration 29
Table 5-1: Summary of all the dataset generated by the MATLAB-script. 30
Table 5-2: Exact parameter-numbers as seen in Figure 5-1 31
Table 5-3: Exact parameter numbers as seen in Figure 5-2 32
Table 5-4: Exact parameter numbers as seen in figure 5-3 33
Table 5-5: Table comparing relative permeability changes in 5.1.1, 5.1.2 and 5.1.3 34
Table 5-6: Exact parameter numbers as seen in figure 5-4 35
Table 5-7: Exact parameter numbers as seen in Figure 5-5 37
Table 5-8: Exact parameter numbers as seen in figure 5-6 38
Table 5-9: A comparison of the dataset seen in figure 5-7 and figure 5-8..... 41
Table 5-10: Absolute permeabilizes with no salt-precipitation in a turbulent flow regime..... 42
Table 5-11: Absolute permeabilities with no-salt precipitation in a laminar flow regime. 42
Table 5-12: Table is presenting the differences in two datasets at different porosities. 43

List of Figures

Figure 3-1: Pore-chamber connected by pore-throats with expected area D 12
Figure 3-2: A distribution curve showing a probability density function 12
Figure 4-1: Bundle of tubes with precipitated salt. 14
Figure 4-2: A single tube in the core precipitated with salt. 15
Figure 4-3: Figure showing the radii of area A and area B. 23
Figure 4-4: Cross section of the core showing how the area A and area B is divided. 24
Figure 4-5: Laminar fluid profile hitting the core. 25
Figure 4-6: Bell Curve showing particles tendency to “random walk” 26
Figure 4-7: A flow chart of the MATLAB file “TurbulentNPart.m” 27
Figure 4-8: A flow chart of the MATLAB file “LaminarNPart.m” 28
Figure 5-1: Dataset for a turbulent flow, porosity of 0.100 30
Figure 5-2: Dataset for a turbulent flow, porosity of 0.184 32
Figure 5-3: Dataset for a turbulent flow, porosity of 0.300 33
Figure 5-4: Plotting the data seen in Table 5-5 34
Figure 5-5: Dataset for a laminar flow, porosity of 0.100 35
Figure 5-6: Dataset for a laminar flow, porosity of 0.184 37
Figure 5-7: Dataset for a laminar flow, porosity of 0.300 38
Figure 5-8: PDF function showing how tubes are distributed in the model 39
Figure 5-9: Same dataset as seen in figure 5-5 40
Figure 5-10: Dataset for turbulent flow, porosity of 0.184 and particle size 15 µm 40

Nomenclature

Abbreviations

CCS: Carbon Capture & Storage

CCSA: Carbon Capture & Storage Association

GHG: Greenhouse gas

IAE: The International Energy Agency

IPPC: The Intergovernmental Panel on Climate Change

PDF: Probability Density Function

Symbols

Theory - parameters

$\bar{\lambda}$: Wavelength of dispersion force

μ : Dynamic viscosity of the fluid

D : Diffusion force

d : Particle radius

D_g : Grain diameter

D_p : Particle diameter

F_n : Constant used if $\left(\frac{s-2}{\bar{\lambda}}\right)$>1

F_R : Repulsive force

F_{VW} : Electromagnetic force

k : Boltzmann's constant, $1.38*10^{-23}$

k : Debye reiporcal double-layer thickness

N_c : Centrifugal number

N_g : Gravity number

N_i : Inertia number

N_{pe} : Peclet number

N_{Re} : Reynolds number

R : Radius of curvature

s : Seperation distance

T : Temperatue, Kelvin

v_a : Velocity of carrier fluid

v_s : Velocity of a spherical particle

w : Angular velocity

ρ : Density of carrier fluid, kg / m^3

ρ_s : Density of particle

τ : Shear Force

Model - parameters

$\tilde{A}$ / $\tilde{B}$ / $\tilde{C}$: Constants

Δp : Net pressure drop across the core, Pa

Δp_1 : Pressure drop across L_1, Pa

Δp_2 : Pressure drop across L_2, Pa

Δr : Radius reduced by salt, m

μ : Dynamic viscosity of the fluid, Pa∗s

μ : Location parameter

A : Area of core, m^2

D_{aq} : Density of aqueous phase, s.g.

D_s : Density of salt, s.g.

I : Injection rate

k_{abs} : Absolute pereability, D / mD

L : Length of the core, m

L_1 : Length of dry-out zone, m

L_2 : Length of uncontaminated zone, m

m : Mean value

m_s : Mass of salt, kg

N : Number of tubes in the core

P : Probability associated with probability density functions (PDF)

Q : Net flow rate across the core, m^3/s

q_i : Flow rate for a single tube, m^3/s

R : Radius of the core, m

R_A: Radius of area A in the core, m

R_B: Radius of area B in the core, m

r_i : Radius of a tube in the core, m

S_s : Solid salt saturation

v : Variance

V_b : Bulk volume, m^3

V_p : Pore volume, m^3

V_s : volume of salt in the core, m^3

X_s : Mass fraction of salt in aqueous phase

α : Fraction of dry-out zone

β : Injectivity rate

σ : Scale parameter

Φ : Porosity

1 Introduction

The climate has been discussed a lot the latest years. We have extracted fossil fuel to reach the ever-increasing energy demand we have worldwide. This has led to increasing CO_2 emission, by using the acquired fossil resources. CO_2 is considered one of the greenhouse-gases (GHG), and scientist has agreed upon that it is better to store it in a geological formation, rather emitting it into the atmosphere.

Carbon dioxide has a longer retention time in the atmosphere. Other greenhouse gases, like for instance methane and nitrous oxide, has effect over the next few decades to centuries. Carbon dioxide stays in the atmosphere for longer and should be considered as a more long-term threat (Solomon, Plattner, Knutti, & Friedlingstein, 2009).

CCS is a key technology for tackling CO_2 emissions. It's affordable, and CCSA has estimated that they will reuce the worlds CO_2 emissions by 19% by 2050. Without CCS the handling of carbon dioxide would be 70% more costly (CCSA, 2011-2017c). Some challenges for CCS are funding long-term, developing frameworks regarding transport and storage of CO_2 and generally getting the public understanding and acceptance for their technology (Gibbins & Chalmers, 2008).

For carbon dioxide to be stored at a geological site, it is not enough to only have a high enough storage capacity. We need two more things; high sustainable injectivity and the ability to get a safe containment. The problem with injectivity to a well, is that we need to know the reservoirs ability to accept carbon dioxide at a high enough rate without making the reservoir lose it's integrity (Birkholzer, Oldenburg, & Zhou, 2015; Schembre-McCabe, Kamath, & Gurton, 2007)

This master thesis should help understanding how a formation may act during injection. A model will be introduced, and it takes into account how colloidal particles and salt precipitation impair the rock.

2 Problem Definition and Objectives

The objective of this master thesis is to investigate CO_2 injection further by developing a model. We will look at two impairments that can happen during CO_2 injection into a rock-formation, and we will look at salt-precipitation and fines migration. Data from lab is used, where CO_2 is injected into a core sample and brine is present to introduce the possibility of salt precipitation.

From the model, we will look at data on mainly; absolute permeability and relative injectivity. The model will be statistical, and we will look at how our main parameters are changed by changing parameters like porosity and salt saturation.

3 Theory

3.1 Climate Change and CCS

Majority of scientist and governments worldwide is agreeing that a climate change is occurring. The main cause is fossil fuels, and it is also agreed upon that storing it is safer than emitting it. Developing countries need fossil fuel to deliver the growing energy demand. If the developing countries must maintain low carbon emissions, they will require flexible developments until more renewables become relevant.

IEA (The International Energy Agency) (CCSA, 2011-2017b), has estimated that by 2030 we need to increase our energy output by 45 percent to meet the worldwide energy demand. To achieve this, we need to use fossil fuels, since it delivers a big fraction of the energy required worldwide.

IPPC (The Intergovernmental Panel on Climate Change) (CCSA, 2011-2017b), found that if we are going to have a reasonable chance for out average global temperature not increase past pre-industrial levels by more than $2^{o}C$, we need to reduce our CO_2 emissions by more than 50-80 percent by 2050. If we are to achieve this, we need better low-carbon technology at rate and scale that we currently have available.

CCS is known as Carbon Capture and Storage. CCS goal is to allow us to continue the use of fossil fuels. It is a technology which captures carbon dioxide (CO_2) from industrial settlements and preventing it from contributing to climate change. CCS is divided into 3 stages; capture, transport and storage.

In the capture stage the carbon dioxide is removed or separated. The sources of the CO_2 can be coal and gas power plants, and plants that manufacture steel and cement. The capture stage of CCS can be divided into three types; post-combustion, pre-combustion and oxyfuel combustion.

After the capture stage, the carbon dioxide is compressed and transported to a storage site. The carbon dioxide can be transported, usually, by pipeline. If the transport must be done offshore, we can also transport by ship.

When the carbon dioxide has been transported, it will often be injected deep underground to a suitable storage location. These storage locations must be a geological site to ensure that the carbon dioxide is stored safe and permanent. The storage in its self can be done within depleted oil and gas fields or deep saline formations. (CCSA, 2011-2017a)

3.2 General CCS Challenges

CCS is considered to be feasible at a commercial scale. Many different kinds of technologies could be used, and currently no scientific breakthroughs are required. There are mainly two reason for why many CCS-projects has yet to be confirmed:

1. Sufficient funding is required, and it has to be long-term.
2. Frameworks has to be set up regarding the transport and geological storage of CO_2

Also, a minor barrier for CCS is that they need to develop public understanding and acceptance (Gibbins & Chalmers, 2008).

In R&D there are certain challenges that has to be identified. Reduction of cost, especially druing the capture stage, is especially important for the CCS. Other facts that also are important are the identification, performance and monitoring at the carbon dioxide storage sites. Even though CCS-technology may be advanced, they will have to consider if the cost is justified to reduce CO_2 emissions for a certain project. The CO_2 reduction needs to bring value according to cost. Legal and regulatory framework for emissions accounting and trading will govern if a emission reduction is viable (Gibbins & Chalmers, 2008).

Regarding the storage stage of CCS, there may be rapid leadkage paths, which is the most common cases are failed wells. If such a problem would occur, they may be redeemed quickly, but even low rates of seepage from the storage site may cause problems. Low rates of seepage, even 0.1% of stored volume, may lead to an increased CO_2 concentration in the athmossphere compared to other instances where this doesn't happen. Important technologies for geological storage are: directional and horizontal drilling for cost-efficient injection, modelling to better understand the injection processes, seismic techniques for locating CO_2 and borehole logging/smart logging for seepage detection (Gibbins & Chalmers, 2008).

3.3 Mechanisms of CO_2 Injectivity Impairment

Salt precipitation during CO_2 injection and Fines migration are two types of impairments that can occur during CO_2 injection. The two are discussed in 3.3.1 and 3.3.2.

3.3.1 Salt precipitation during CO_2 injection

Injecting CO_2 into a formation containing saline water may lead to severe injectivity decline due to salt precipitation (Grude, Landrø, & Dvorkin, 2014; Muller, Qi, Mackie, Pruess, & Blunt, 2009; Ott, Roels, & De Kloe, 2015; Peysson, Andre, & Azaroual, 2014). Formation water will start evaporating when dry CO_2 is injected into the reservoir. This leads to a higher mole fraction of water in the CO_2 stream, and increased concentration of dissolved salt in the formation water. At one point the salt concentration will exceed the solubility limit at the given conditions in the reservoir, and solid salt will start precipitating and alter the porosity and permeability of the formation. (Miri & Hellevang, 2016)

Evaporation of trapped water in the porous media may also increase the relative permeability of CO_2 by providing more space for CO_2 to flow. This must not be confused with the absolute permeability, which can still be reduced as the relative permeability is increased. It is important to take both the effects into consideration to get a realistic view of how the injectivity changes (Ott et al., 2015; Roels, Ott, & Zitha, 2014).

There are several physical mechanisms that have been identified to govern the dry-out and precipitation of salts. These mechanisms are (Miri & Hellevang, 2016):

1. "Two-phase displacement of brine away from the injection well by viscous pressure gradients imposed through injected CO_2"
2. "Evaporation of brine into the flowing CO_2 stream"
3. "Capillary-driven back-flow of aqueous phase toward the injection point due to capillary pressure gradients"
4. "Molecular diffusion of dissolved salt in the aqueous phase"
5. "Gravity override of injected CO_2"

"Salt self-enhancing"

The first CO_2 that is injected into the reservoir will physically displace saline water already present in the formation. This results in a to phase flow zone with a water phase and a CO_2 phase. Some residual brine will be left behind in the formation after the majority has been

displaced (Ott et al., 2015; Peysson et al., 2014; Pruess & Müller, 2009). Residual brine can typically be found in as a wetting film around the grains, in small porous spaces and porous spaces with little to no communication (Miri, van Noort, Aagaard, & Hellevang, 2015). These areas are now exposed to a flow of dry CO_2, initiating the evaporation of the residual brine. This results in drying out the area closest to the injection point. This creates a high salt saturation gradient in the evaporating front, resulting in capillary pressure gradient building up (Peysson et al., 2014). The capillary pressure gradient will eventually exceed the injection pressure gradient in the region, and more brine will be transported to the evaporating front (Ott et al., 2015; Peysson et al., 2014; Pruess & Müller, 2009). Water will be dissolved in the CO_2 and salt concentration in the trapped water will increase and lead to salt diffusion (Pruess, 2009; Shahidzadeh-Bonn, Rafaï, Bonn, & Wegdam, 2008).

Salt will precipitate when it reaches the solubility limit at the given thermodynamic conditions. The salt is very water wet, and transports more water to the evaporation front, increasing the precipitation further (Miri et al., 2015). The capillary forces due to the salt is much stronger than the capillary pressure due to different salt concentration, but both effects leads to capillary backflow to the evaporation front (Miri & Hellevang, 2016).

3.3.1.1 Drying regimes

Salt precipitation can either occur local or non-local, dependent on the active drying regime. Three drying regimes have been identified based on experiments and numerical modeling. (Miri & Hellevang, 2016).

3.3.1.1.1 Diffusive regime

The diffusive regime is active during low injection rates of CO_2, and results in a low evaporation rate. The evaporation induces a capillary pressure gradient that transports more saline water to the injection point (Ott, Snippe, De Kloe, Husain, & Abri, 2013; Peysson et al., 2014). This results in scenario where the evaporation rate is the same as the capillary backflow, preventing the formation of a drying front (Andre, Peysson, & Azaroual, 2014; Peysson et al., 2014). Water evaporating increases the salt super-saturation, and salt diffusion is the dominating depositing mechanism (Peysson, 2012; Peysson et al., 2014). This gives a more homogeneous distribution of salt in the reservoir formation.

3.3.1.1.2 Capillary regime

The capillary regime is active when the initial evaporation is greater than the capillary backflow, but reaches equilibrium later. The area closest to the injection point will become dry, and the drying front will advance until the capillary backflow equals the evaporation (Kim, Han, Oh, Kim, & Kim, 2012; Peysson et al., 2014; Pruess & Müller, 2009). The initial evaporation is high, and the salt is precipitated much faster than it is diffused from the drying front. The capillary regime results in a massive salt accumulation at the drying front, but distributes evenly in the dry out zone behind the drying front. Some experiments have shown that the salt accumulated in the capillary regime have a porous and permeable structure (Miri et al., 2015).

3.3.1.1.3 Evaporative regime

Evaporative regime is active when the injection rate is high, and is over the critical limit (Andre et al., 2014; Giorgis, Carpita, & Battistelli, 2007; Kim et al., 2012; Ott, De Kloe, Marcelis, & Makurat, 2011; Ott et al., 2015). The rate of evaporation will always be greater than the capillary backflow, and the drying front continuous to move deeper into the aquifer. The trapped brine and coating is immediately evaporated by the drying front. The salt is thought to distribute homogenously throughout this region. Most models predict a precipitation equal to that of the salt in the coating film and trapped brine (Giorgis et al., 2007; Pruess & Müller, 2009; Zeidouni, Pooladi-Darvish, & Keith, 2009). Some recent research show however that the precipitation could be much larger due to the strong capillary forces created by the already precipitated salt. (Miri & Hellevang)

3.3.1.2 Mitigating actions

The knowledge of the physics and mechanisms behind salt precipitation is very limited, and there exist very few mitigating actions. The most popular is injecting freshwater to dissolve already precipitated salt, and transport it further into the formation. Freshwater injection can also be done prior to CO_2 injection to reduce the salinity of the near wellbore fluid.

Salt precipitation is a major problem for CO_2-injections, but little success has been made to quantify the effect it will have on the injectivity. The uncertainty of the injectivity decline that is caused by salt precipitation is still high, and further research and new numerical models are necessary (Miri & Hellevang, 2016).

3.3.2 Fines Migration

Small particles in porous media can be called fine particles or fines. Migration of fine particles is a problem, important for both scientifically and industrially. Fines migration means, in this case, the entire process of where a particle is released, taken by the flow and then captured by the pores in the porous medium. The detachment of the fine particles is known to be caused by colloidal and hydrodynamic forces.

When a fine particle gets captured by a constriction, mainly two things can happen. In the first case, the fine particle may get captures by the constriction, resulting in reduced flow through the porous medium (Khilar & Fogler, 1998). In the second case, the particle may not the captured by the constriction, but rather just erode the medium and move on. Both cases may be desirable in their own way, so techniques may be introduced to induce or prevent fines migration.

Forces acting on a particle in flowing suspension can be classified in three categories. Forces related to- transport mechanism, attachment mechanism, and detachment. (Ives, 1985).

3.3.2.1 Forces that relates to transport mechanisms

The relevant quantities under this chapter are as follows: d is the particle diameter, D is the porous grain diameter, ρ_s is the density of the particles, ρ is the density of the carrier fluid and μ is the viscosity of the carrier fluid, v_a is the convective velocity, g is the acceleration of gravity and T is the absolute temperature.

3.3.2.1.1 Inertia force

Inertia is the force that gives the particle the ability to maintain its momentum travelling in a straight line. It can be expressed as (Ives, 1985)

$$N_i = \frac{\rho_s D_p^2 v_a}{18 \mu D_g} \qquad (3.1)$$

3.3.2.1.2 Gravity force

There may be a density difference between the carrier fluid and the particle. The particles will then move in the gravity direction according to Stokes' law. The velocity is given by

$$v_s = \frac{(\rho_s - \rho) g D_p^2}{18\mu} \quad (3.2)$$

Particles will become buoyant when they are lighter than the carrier fluid, the gravity force will react upwards. The opposite is true for a heavier particle, making the particles settle. We can express the gravity force in a dimensionless group called gravity number (Ives, 1985)

$$N_g = \frac{g(\rho_s - \rho) D_p^2}{18\mu v_a} \quad (3.3)$$

3.3.2.1.3 Centrifugal forces

External acceleration will generate centrifugal forces. An angular velocity w and radius R will create centrifugal force. The Centrifugal force can be expressed in following dimensionless form:

$$N_c = \frac{Rw^2(\rho_s - \rho) D_p^2}{18\mu v_a} \quad (3.4)$$

3.3.2.1.4 Diffusion force

A particle smaller than 1.0mm tends to move irregularly when it is dispersed in a liquid. This is a phenomenon called Brownian motion. Particles undergoing Brownian movements is expressed by Einstein (McDowell-Boyer, Hunt, & Sitar, 1986):

$$D = \frac{kT}{3\pi\mu D_p} \quad (3.5)$$

Where k = 1.38 x 10^{-23} is the Boltzmann's constant. The diffusion force can also be expressed by Peclet number. The ratio of convection velocity to average Brownian velocity is given by (Ives, 1985)

$$N_{pe} = \frac{D_g v_a}{D} = \frac{3\pi\mu D_p D_g v_a}{kT} \quad (3.6)$$

3.3.2.1.5 Hydrodynamic force

Fluid shearing and pressure forces make up the Hydrodynamic forces (A. Wojtanowicz, Krilov, & Langlinais, 1987). The motion of the fluids will move the fine particles along. The hydrodynamic force can be expressed by the dimensionless group given by Reynolds number

$$N_{Re} = \frac{v_a D_g \rho}{\mu} \quad (3.7)$$

When the particles are small, the fluid velocity can be equal to the particle velocity.

3.3.2.2 Forces that relate to attachment mechanisms

Forces from attachment mechanisms can act upon a particle when it closer than 1μm away from the grain surface. (Ives, 1985)

3.3.2.2.1 London-van der Waals force

Electronic characteristic of atoms and molecules generate electromagnetic waves, making them attract to each other. This electromagnetic force can be expressed as

$$F_{VW}(s) = \frac{1}{(s-2)^2} F_n\left(\frac{s-2}{\bar{\lambda}}\right) \quad (3.8)$$

Where $\bar{\lambda}$ is a dimensionless wavelength of dispersion force, s is the dimensionless separation distance, and F_n is used according to if $\left(\frac{s-2}{\bar{\lambda}}\right)$ is less or greater than 1.

3.3.2.2.2 Friction-drag force and hydrodynamic thinning

When a particle is about to attach on the grain surface, the particles will experience a friction force because they need to displace the fluid that is already present in the grain surface (Ives, 1985; Khilar & Fogler, 1998)

3.3.2.3 Forces that relates to detachment mechanisms

3.3.2.3.1 Shearing force

A particle may become detached and mobilized when the shear-force of the passing fluid is greater than the forces attaching the particle to the grain surface (Ives, 1985).

$$\tau = \mu \frac{dv}{dr} \quad (3.9)$$

3.3.2.3.2 Electrostatic Double-Layer Force

These forces are created by ionic conditions. If the particle and the grain carry the same charge, then they would repel each other. The repulsive force can be expressed as (Ives, 1985):

$$F_R(s) = \frac{e^{[-kd(s-2)]}}{1 + e^{[-kd(s-2)]}} \quad (3.10)$$

Where s is the ratio between separation distance and the particle radius, k is the Debye reciprocal double-layer thickness, and d is the particle diameter.

3.3.2.3.3 Born Repulsion Force

When electron clouds are overlapping, a detachment force may occur. If, let's say two clouds, where to overlap, the cluster of negatively charged electrons would create this repulsion force (K. A. Wojtanowicz & Krilov, 1988).

3.4 Pore-Scale Modelling

Inside a porous material we find void spaces, which are free of solids. These void spaces are generally known as pore space. For a fluid to permeate through a porous medium, the pore space must be continuous (Scheidegger, 1974). A pore will have to be some sort of shape, so that it has the chance to be interconnected with other pores. Pores can also be non-interconnected, and will become dead-end pores (Dullien, 2012; Scheidegger, 1974).

Effective pore space, is the space that is interconnected. This makes it possible to transport matter (Lymberopoulos & Payatakes, 1992). One of the easiest ways to concept how a pore space could like is to stack spherical grains. The most stable rhombohedral packing would have a porosity, φ, of 0.2595. With this kind of packing we will get pore chambers and pore constrictions. These pore constrictions will be connected in more than one direction (Khilar & Fogler, 1998).

Coordination number is generally the way to characterize connectivity of pore networks. The Coordination number represent the number of independent paths between the pore chambers. We want to have a number for how many pore chambers that are connected to a pore constriction (Khilar & Fogler, 1998). For sandstones an average coordination number is between 4 and 8 (Lin & Cohen, 1982). For a rock with higher coordination number, it will be less likely for the all the constrictions to be plugged (Khilar & Fogler, 1998). We can show how pore throats are connected, in sandstones, by Figure 3-1. Here we have four pore-throats connected to a single pore-chamber. This is only a picture in the 2D-plane, so the number coordination number could be higher than 4. In the 3D-plane, we could for instance have two extra pore-throats; one going "into the paper", and one coming "out the paper". In this case, the coordination number would be 6.

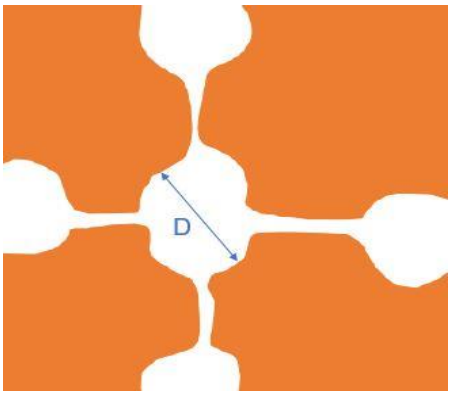

Figure 3-1: Pore-chamber connected by pore-throats with expected area D

If we are going to make a model out of this, we have to take the coordination number into consideration. For a model with a bundle of tubes (parallell tubings with non-interconnectivity shown in Figure 4-1), instead of pore- chambers and throats, we would have to use a stasticial curve that takes into consideration the varying sizes of tubes. Some of the tubes could have a higher diameter than the expected size, and some could a lower diameter. Taken into consideration the pore-throats, it would be justified to say that the size-distribution should have a higher concentration of lower diameter-tubes, rather than higher diameter-tubes. An example of this distribution is shown in Figure 3-2 below.

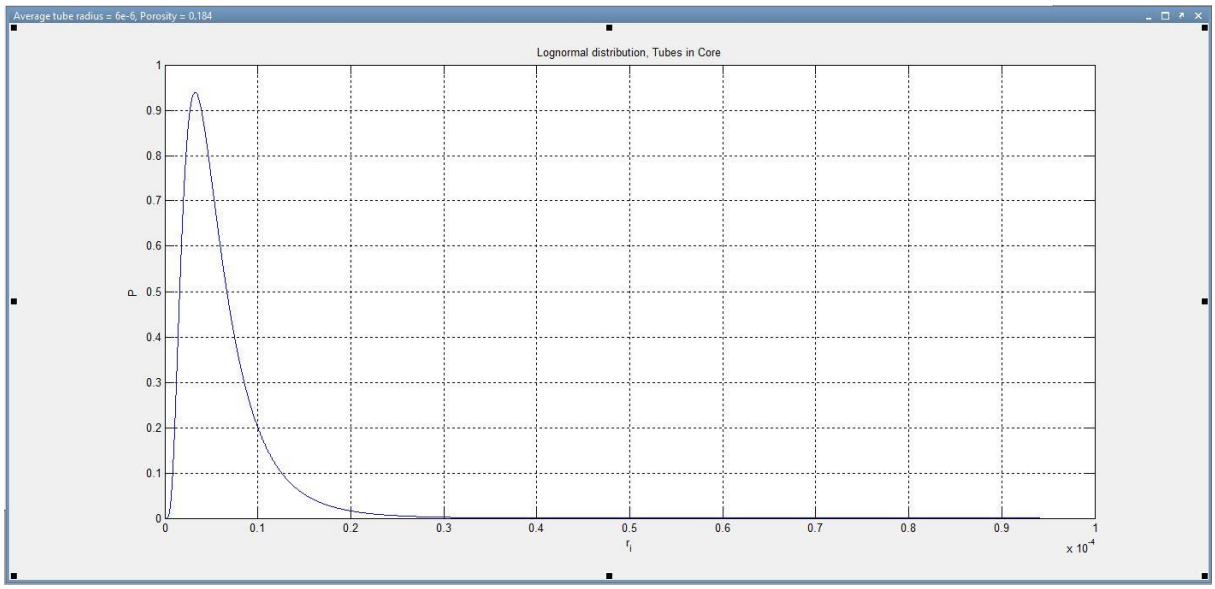

Figure 3-2: A distribution curve showing a probability density function

3.5 Flow patterns

3.5.1 Laminar

Laminar flow, also called streamline flow, is when fluid flows in parallell layers with no disruption between them (Batchelor, 2000). The movement in laminar flows are very orderly and the layers are moving in straight lines, which are parallell to the surfaces (Cath & Andrew, 2009). When considering viscous fluid through a pipe, we will get a flow pattern where the veolocity of the fluid near the wall is zero and increasing towards a maximum near the cross-sectional centre of the pipe (Nave, 2005). An example of this kind of flow, is further discussed in section 4.3.1.

Reynolds number, shown in (3.7), is an important parameter to figure our if a flow should be laminar or turbulent (described in 3.5.2). Laminar flow will occur when the Reynolds number is below the critical value of about 2040. However, the transition range is considered to between 1800 and 2100 (Avila et al., 2011).

3.5.2 Turbulent

Due to chaotic changes in pressure and fluid velocity, turbulent flows will occur (Batchelor, 2000). Turbulent flows creates unsteady vortices. These vortices appear in many sizes, and will interact with each other to create drag due to friction. This will in turn increase the energy we require to pump a turbulent through a pipe, compared to a laminar flow. Turbulent flows will occur when the Reynolds number is above the critical value of about 2040 (Avila et al., 2011).

3.5.3 Brownian movements

When a particle is submerged into a fluid, fast moving atoms or molecules will collide with the particle. These collisions will result in the random motion called Brownian movements (Richard, 1970). Robert Brown was the one to observe this transport phenomenon in 1827. He observed, through a microscope, that particles moved randomly throughout water. He wasn’t able to figure out the mechanisms of these movements. Albert Einstein then published a paper in 1905, which explained that the particles had been moved by individual water molecules (Wikipedia, 2017).

4 Model Development

4.1 Fundamental Assumptions and Simplifications

4.1.1 Overview

The model used here is a simplified model which uses a bundle of tubes, rather than a complex system of pore-throats and channels. The porous medium will be separated into a parallel system of tubes.

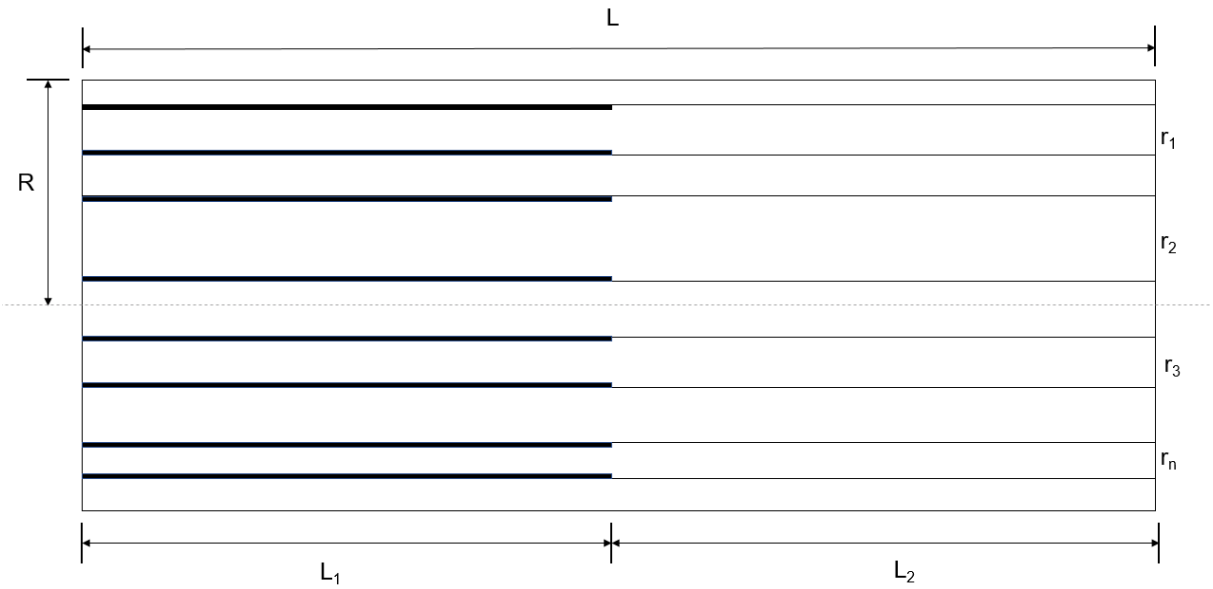

Figure 4-1: Bundle of tubes with precipitated salt.

In Figure 4-1, L describes the entire length of the core, L_1 is the dry-out zone (where salt has precipitated), L_2 is the uncontaminated zone. R is the radius of the entire core and (r_1, r_2, r_3, …, r_n) is the varying lengths of each tube-radii.

For a single tube in the core we can define the reduction of radii as Δr_i according to which tube we are looking at. The dry-out zone, L_1, is defined to be the same for each tube in the core.

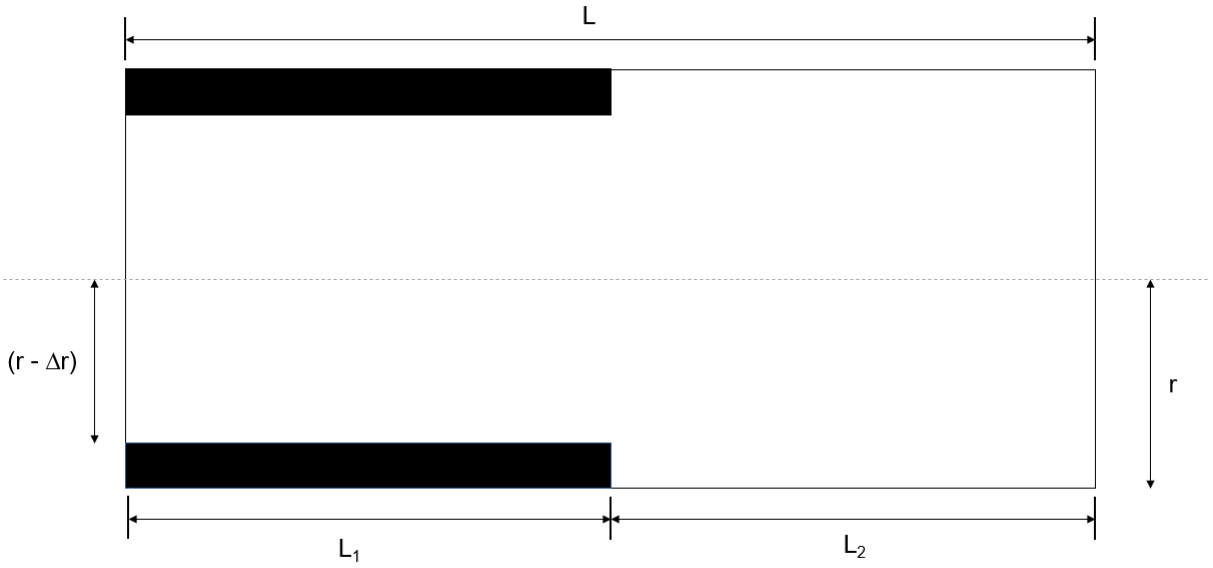

Figure 4-2: A single tube in the core precipitated with salt.

We introduce a dry-out coefficient, α, to define a ratio between dry-out zone and the length of the core:

$$\alpha = \frac{L_1}{L} \tag{4.1}$$

We restructure equation (4.1) to express dry-out zone, L_1, as:

$$L_1 = \alpha L \tag{4.2}$$

The uncontaminated zone, L_2, will then be expressed as:

$$L_2 = (1 - \alpha)L \tag{4.3}$$

4.1.2 Fluid flow through a single tube

From Figure 4-2 we assume that the dry-out zone and the uncontaminated zone can be connected as two different radii tubes. The net pressure drop across these tubes can be expressed as

$$\Delta p = \Delta p_1 + \Delta p_2 \tag{4.4}$$

Where Δp_1 and Δp_1 are, the pressure drops of the dry-out zone and the uncontaminated zone respectively. Using Poiseuille's law, Δp_1 and Δp_1 can be expressed as

$$\Delta p_1 = \frac{8 q_1 \mu L_1}{\pi (r^4 - \Delta r)^4} \tag{4.5}$$

$$\Delta p_2 = \frac{8 q_2 \mu L_2}{\pi r^4} \tag{4.6}$$

Where, q_1 and q_2 are the flow rate of the fluid across dry-out- and the uncontaminated zone respectively. μ is the dynamic fluid viscosity. We do neither gain nor lose our flow, so we can assume that

$$q_1 = q_2 = q \tag{4.7}$$

Now we can substitute equation (4.2), (4.3), (4.5), (4.6) into (4.4) to find the pressure drop across a singular tube in our core

$$\Delta p = \frac{8 q \mu L}{\pi} \left[\frac{\alpha}{(r - \Delta r)^4} + \frac{(1 - \alpha)}{r^4} \right] \tag{4.8}$$

4.1.3 Fluid flow through N amount of tubes

Since we look at our core as a bundle of cylindrical parallel tubes, the fluid rate through the core will look something like

$$Q = q_1 + q_2 + q_3 + \cdots + q_n = \sum_{i=1}^{N} q_i \tag{4.9}$$

The pressure drop is the across each tube

$$\Delta p = \Delta p_1 = \Delta p_2 = \Delta p_3 = \cdots = \Delta p_N \tag{4.10}$$

If we combine equation (4.8), (4.9) and (4.10) , we can express the total flow rate as

$$Q = \frac{\pi \Delta p}{8 \mu L} \sum_{i=1}^{N} \left[\frac{(r_i - \Delta r)^4}{\alpha + (1 - \alpha)(1 - \frac{\Delta r_i}{r_i})^4} \right] \tag{4.11}$$

4.1.4 Injectivety impairment induced by salt precipitation

Fluid injectivety is defines as ratio of the injection flow rate to the pressure drop

$$I = \frac{Q}{\Delta p} \tag{4.12}$$

We can combine equation (4.11) and (4.12) to express fluid injectivety as

$$I = \frac{\pi}{8\mu L}\sum_{i=1}^{N}\left[\frac{(r_i - \Delta r)^4}{\alpha + (1-\alpha)(1-\frac{\Delta r_i}{r_i})^4}\right] \tag{4.13}$$

Relative injectivety change β is introduced to quantify the effect the effect of salt precipitation on CO_2 injectivety

$$\beta = \frac{I_i - I_f}{I_i} = 1 - \frac{I_f}{I_i} \tag{4.14}$$

Where I_i and l_f is the fluid injectivety before and after salt precipitation respectively.

Before salt precipitation Δr and α will be 0. Inserting equation (4.13) into (4.14) yields

$$\beta = 1 - \frac{I_f}{I_i} = 1 - \frac{\sum_{i=1}^{N}\left[\frac{(r_i - \Delta r)^4}{\alpha + (1-\alpha)(1-\frac{\Delta r_i}{r_i})^4}\right]}{\sum_{i=1}^{N} r_i^4} \tag{4.15}$$

To solve equation (4.15), we need to find Δr, α, N and r_i for every tube

4.1.5 Total number of capillary tubes, N

Porosity φ is defined as the ratio between the pore volume V_p and bulk volume V_b.

$$\phi = \frac{V_p}{V_b} \tag{4.16}$$

To find the pore volume V_p, we can sum the internal volume of all the tubes.

$$V_p = \pi L \sum_{i=1}^{N} r_i^2 \approx \pi L N \overline{r_i^2} \tag{4.17}$$

Where $\overline{r_i^2}$ is the average value of the square of the tube radii.

The bulk volume of the core can be expressed as

$$V_b = \pi R^2 L \tag{4.18}$$

Substituting equation (4.17) and (4.18) into (4.16) gives us

$$N = \phi \frac{R^2}{\overline{r_i^2}} \tag{4.19}$$

We can show, by integration

$$\overline{r_i^2} = \frac{1}{\Delta r_{max}} \int_0^{\Delta r_{max}} r_i^2 dr_i \tag{4.20}$$

By doing the integration in equation (4.20), we can show that

$$\overline{r_i^2} = \frac{4}{3} \overline{r_i}^2 \tag{4.21}$$

Where $\overline{r_i}^2$ is the square of the average tube radii. We can then substitute equation (4.21) into (4.19), and find the total number of tubes in the core

$$N = \frac{3}{4} \phi \left(\frac{R}{\overline{r_i}}\right)^2 \tag{4.22}$$

Now we can estimate the total number of N tubes if we know the porosity, φ, and the average tube radius, $\overline{r_i}^2$.

4.1.6 Thickness of precipitated salt, Δr

The solid salt saturation in a single tube is defined by

$$S_{si} = \frac{V_{si}}{V_p} \tag{4.23}$$

V_{si} is the volume of precipitated salt in the tube. From Figure 4-2 we can estimate the volume of salt (shaded area) as

$$V_{si} = \pi L \alpha (2\Delta r_i r_e + \Delta r_i^2) \tag{4.24}$$

Where $r_e = (r_i - \Delta r_i)$. The thickness of precipitated salt can be expected to be very small, and then probably fall within 10^{-7} to 10^{-9} region in meters. This means Δr_i will be small, and Δr_i^2 will be neglectable. We can rewrite equation (4.24) as

$$V_{si} = \pi L \alpha (2\Delta r_i r_e) \tag{4.25}$$

The pore volume can be defined as the volume of the tube, and we can substitute equation (4.25) into (4.23) and get

$$S_{si} = \frac{\pi L \alpha (2\Delta r_i r_e)}{r_i^2} \quad (4.26)$$

Substituting, $r_e = (r_i - \Delta r_i)$ into equation (4.26), we will get

$$S_{si} = 2\alpha \Delta r_i (\frac{1}{r_i} - \frac{\Delta r_i}{r_i^2}) \quad (4.27)$$

We can assume $\frac{\Delta r_i}{r_i^2}$ = 0, because $r_i \gg \Delta r_i$.

Now we can express the precipitated salt in a tube as

$$\Delta r_i = \frac{S_{si} r_i}{2\alpha} \quad (4.28)$$

We can estimate mass of precipitated salt in the tube

$$m_{si} = \rho_s \pi \alpha L 2 r_i \Delta r_i \quad (4.29)$$

Where ρ_s is the density of precipitated salt.

Total mass of mass of precipitated salt can be estimated as

$$m_t = \sum_{i=1}^{N} m_{si} \approx N\overline{m_{si}} \quad (4.30)$$

We can also define the total mass of salt, m_t by

$$m_t = \rho_s V_{st} \quad (4.31)$$

Where V_{st} is the total volume of salt in the tubes within the core. V_{st} can further be expressed as

$$V_{st} = S_s \pi R^2 L \phi \quad (4.32)$$

Where S_s is the total salt saturation of the core.

If we substitute equations (4.22), (4.29), (4.31), (4.32) into (4.30), we will get

$$\Delta r_i = \frac{2}{3} \frac{S_s \bar{r}_i}{\propto} \quad (4.33)$$

With equation (4.33) we express the average solid salt saturation in terms of total precipitated salt in each tube

4.1.7 Absolute permeability

Absolute permeability $k_{absolute}$ of porous rock is defined in Darcy's equation (Darcy, 1856):

$$Q = -k_{absolute}\frac{A}{\mu}\frac{dP}{dx} \tag{4.34}$$

Q is the volume flux through a porous medium; A is cross sectional area of the sample; μ is the dynamic viscosity of the fluid used; dP / dx is the pressure drop across the length of the medium used.

We can describe the equation of laminar viscous flow in a pipe by

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} = \frac{1}{\mu}\frac{dP}{dx} \tag{4.35}$$

where u is the velocity of the fluid; μ is the dynamic viscosity; dP / dx is the pressure gradient; r is the radius coordinate and x is the axial coordinate.

We can find a general solution of equation (4.35)

$$u = \tilde{A} + \tilde{B}r^2 + \tilde{C}lnr \tag{4.36}$$

Where $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ are constants.

We can derive our general solution from equation (4.35)

$$\frac{\partial u}{\partial r} = 2\tilde{B}r + \frac{\tilde{C}}{r}, \qquad \frac{\partial^2 u}{\partial r^2} = 2\tilde{B} - \frac{\tilde{C}}{r^2} \tag{4.37}$$

If we substitute the equations from equation (4.37) into (4.35), we will get

$$2\tilde{B} - \frac{\tilde{C}}{r^2} + 2\tilde{B} + \frac{\tilde{C}}{r^2} = \frac{1}{\mu}\frac{dP}{dy} \tag{4.38}$$

We find a solution for B

$$\tilde{B} = \frac{1}{4\mu}\frac{dP}{dx} \tag{4.39}$$

We need to avoid a singularity when r = 0, because ln (0) will act in-proper. $\tilde{C}$ must be zero to avoid this. We employ no-slip condition u = 0 when r = b.

$$u = \tilde{A} + \tilde{B}r^2 = \tilde{A} + \frac{1}{4\mu}\frac{dP}{dx}r^2 = \tilde{A} + \frac{1}{4\mu}\frac{dP}{dx}b^2 \tag{4.40}$$

We find a solution for A

$$\tilde{A} = -\frac{1}{4\mu}\frac{dP}{dx}b^2 \tag{4.41}$$

Substituting equation (4.41) into (4.40), gives us

$$u = -\frac{1}{4\mu}\frac{dP}{dx}b^2\left(1 - \frac{r^2}{b^2}\right) \tag{4.42}$$

The total volume flux through our pipe can be described as

$$q = -\frac{\pi b^4}{8\mu}\frac{\Delta P}{l} \tag{4.43}$$

We find out absolute volume flow by summing up all of our volume flux' through the pipes

$$Q = \sum_{i=1}^{N} q = -\frac{\pi}{8\mu}\frac{\Delta P}{l}\sum_{i=1}^{N} {r_i}^4 \tag{4.44}$$

Where l is the length of the pipe; ΔP is the pressure difference across the tube lengths; and r_i Is the individual radius a given tube (Dvorkin, 2009)

Substituting equation (4.34) into (4.44) gives us

$$k_{abs} = \frac{\pi}{8A}\sum_{i=1}^{N} {r_i}^4 = \frac{1}{8R^2}\sum_{i=1}^{N} {r_i}^4 \tag{4.45}$$

We can generalize equation (4.45)

$$k_{abs} = \frac{1}{8R^2}\sum_{i=1}^{N} (r_i - \Delta r_i)^4 \tag{4.46}$$

Where Δr_i to expresses how our absolute permeability would change if it was precipitated by salt. If no precipitated salt were present then Δr_i will be zero.

4.2 Overview of equations used in the model

Below are the functions used, and derived, equations used in the MATLAB script to generate the datasets.

$$\Delta r_i = \frac{2}{3}\frac{S_s \bar{r}_i}{\propto} \tag{4.47}$$

Where Δr_i is the amount of blockage from precipitated salt in each tube, S_s is the salt saturation, $\bar{r}_i$ is the average radius of tubing "i", and α is the dry out coefficient. Equation (4.47) is used to generate the radii-reduction of salt precipitation in the MATLAB-script.

$$N = \frac{3}{4}\,\phi\left(\frac{R}{\bar{r}_i}\right)^2 \tag{4.48}$$

Where N is the total number of tubing in the core, ϕ is the porosity and $\bar{r}_i$ is the average tubing radius. Equation (4.48) is used to estimate the amount of tubes in the MATLAB-script.

$$S_s = \left(0.85 + \frac{\propto}{3.5}\right)\left(\frac{D_{aq} * X_s}{D_s}\right) \tag{4.49}$$

Where $\propto$ is the dry out coefficient, X_s is the mass fraction of salt in the aqueous phase, and D_{aq} and D_s is the density of the aqueous and salt respectively. This equation was derived through mass balance and by fitting the experimental data. Equation (4.49) will estimate the salt saturation in the MATLAB-script.

$$k_{abs} = \frac{1}{8R^2}\sum_{i=1}^{N}(r_i - \Delta r_i)^4 \tag{4.50}$$

$$k_{abs,A} = \frac{1}{8R_A^2}\sum_{i=1}^{N}(r_i - \Delta r_i)^4$$

$$k_{abs,B} = \frac{1}{8(R^2 - R_A^2)}\sum_{i=1}^{N}(r_i - \Delta r_i)^4$$

k_{abs} is absolute permeability, R is the total radius of the core sample, N is the total amount of tubes in the core, r_i is the radius of a given tube, Δr_i is a portion of the radius in a tube blocked by salt precipitation.. The difference for $k_{abs,A}$ and $k_{abs,B}$ is due formula (4.45), which consider

area, A. This will make the formula for permeability in area A and area B different. Description of the areas is shown in Figure 4-3. For the MATLAB-script k_{abs} is run for "TurbulentNPart.m", and $k_{abs,A}$ / $k_{abs,B}$ is both run for "LaminarNPart.m".

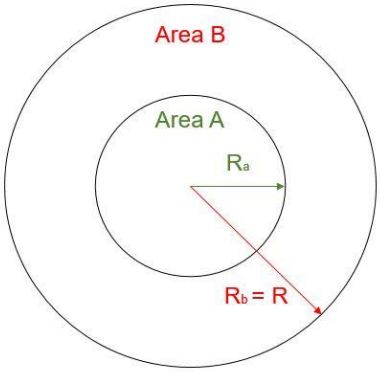

Figure 4-3: Figure showing the radii of area A and area B.

$$\beta = 1 - \frac{k_f}{k_i} \tag{4.51}$$

Where β is the relative permeability/injectivity, k_f is the absolute permeability at a state of our choosing, and k_i is the absolute permeability before salt-precipitation. We can compare relative permeability/injectivity with the impairment-change in percent, as the formula will end up being the same.

$$\mu = ln\left(\frac{m}{\sqrt{1 + \frac{v}{m^2}}}\right) \tag{4.52}$$

Where μ is the location parameter, m is the mean value and v is the variance.

$$\sigma = \sqrt{ln\left(1 + \frac{v}{m^2}\right)} \tag{4.53}$$

Where σ is the scale parameter, m is the mean value and v is the variance.

$$\mathrm{P} = \frac{1}{x\sigma\sqrt{2\pi}} e^{\left[-\frac{(lnx-\mu)^2}{2\sigma^2}\right]} \tag{4.54}$$

This is how the probability density function is defined in MATLAB. We are using a built-in MATLAB-function to use the PDF ("lognrnd"). The probability density function is used to generate a distribution for both tubing- and particle sizes in the MATLAB-script.

4.3 Statistical Modelling of Particle Transport

The core has been split up into two areas, which has been called area A and area B, which is shown in Figure 4-4. Area A has been defined to contain 80% of the total diameter, and thus area B will have 20% of the total diameter. Relative area (in percent) can be expressed as

$$A_{rel,A} = 0.8^2 = 0.64 \tag{4.55}$$

$$A_{rel,B} = 1 - V_{rel,A} = 0.36$$

Area A will contain 64% of the total area, while area B will contain 36% of the total area. Since the volumes are considered relative we can exclude $\frac{\pi}{4}$. When we estimate the total amount of tubes (N) by (4.48, we will then divide the tubes across the two areas

$$N_A = A_{rel,A}N = 0.64N \tag{4.56}$$

$$N_B = A_{rel,B}N = 0.36N$$

For simplicity sake, we assume that any tube has the chance to be anywhere in the cores, and thus having the same distribution in both area A and area B. This means that 64% of the tubes are in area A and the rest of the tubes, which is 36%, is in area B.

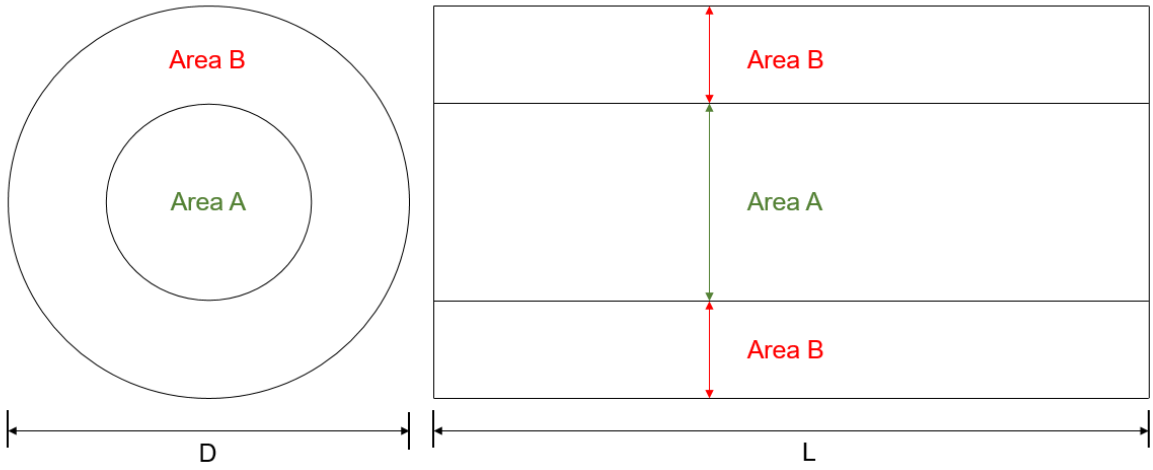

Figure 4-4: Cross section of the core showing how the area A and area B is divided.

4.3.1 Laminar flow estimation

We can consider the flow in the tube acting like a laminar flow, where the flow rate is less against the wall and increases cross sectional center (Nave, 2005)

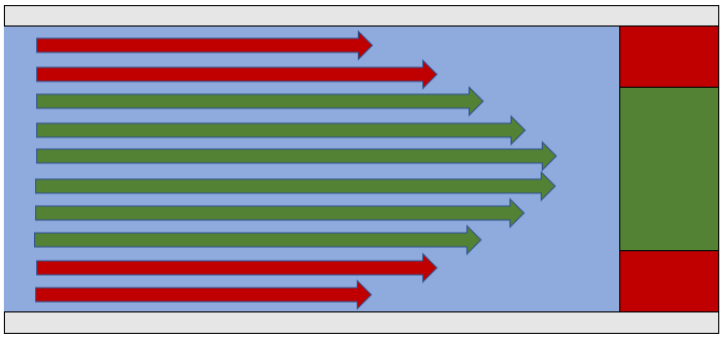

Figure 4-5: Laminar fluid profile hitting the core.

In Figure 4-5 the green arrows are representing the amount of flow hitting the green area of the core, which is representing area A. Likewise, the red arrows will represent the part of the flow that is hitting the red area (area B). In the MATLAB-script, described by section 4.4, we have defined an 80% chance to hit area A and a 20% chance to hit area B. This is just a rough estimation.

4.3.2 Turbulent flow estimation

As discussed in 3.5.2 we can assume that turbulent flows, being totally random, will lead to particles hitting random tubes in the core at this kind of flow.

Another factor for considering where our particles will end up, can be Brownian motions. Einstein worked with Brownian motions and he required a statistical analysis for the random "jiggling" of the particles. If a particle starts from a given start position, then the particle has a chance to migrate further away from the initial point as the time increases. This can be represented by a bell curve. (Norton, 2005) The bell curve shown below, Figure 4-6, consider that the particle starts in its expected starting area, and is moving on its own concordance by being bombarded with gas molecules. Since we have a turbulent flow, the particle can't be standing still. For simplicity sake, we can expect that the particle will appear where it first was estimated to appear, which could be anywhere. For simplicity sake, we can assume that the flow patterns from turbulent flows are the ones that will be valid. This means that it is totally random where a particle will hit the core in the model.

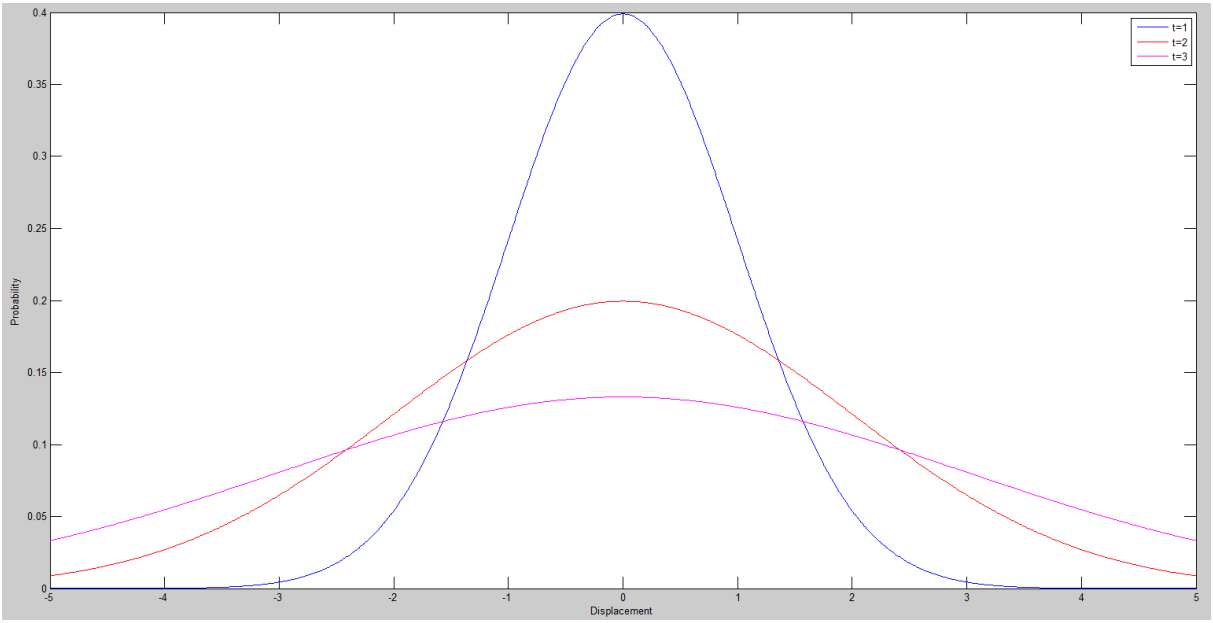

Figure 4-6: Bell Curve showing particles tendency to "random walk"

4.4 Computational Algorithm

The computational algorithm is divided into two parts, where the first one called "TurbulentNPart.m" while the other one is called "LaminarNPart.m". The first gives us the simulations from a situation where the flow is turbulent while the second one gives the situation where the flow is laminar.

4.4.1 MATLAB algorithm "TurbulentNPart.m"

At the start of the script, we are putting all our initial values necessary to estimate the amount of N tubes (equation (4.48)) and initial absolute permeability (equation (4.50)). Afterwards the distribution for the tubing sizes are generated by equation (4.52), (4.53), (4.54).

Next, the salt precipitation is considered. The amount of blockage in the tubes are calculated by equation (4.47). We have two datasets to consider and therefore the script is split into two ("Run 1 – Low salt concentration" and "Run 2 – High salt concentration"). The code is the same. The only difference will be the amount of blockage that is estimated.

The number of particles will be defined and set to zero. At this point the loops will start to run. and a while-statement will check if the particle number is below 1 million. If we have less than 1 million particles, then we will generate a lognormal distribution for the number of particles we have.

At this point the particle number “i” will be going through one random N tube, generated by the “randi” function in the script. The maximum number of particle “i” will change throughout the script and will be (0, 100k, 200k, …, 1M). If particle “i” is smaller than the random tube it goes through, then the tube will not be plugged and nothing will change. However, if particle “i” is bigger than the random tube it goes through, then the tube will be plugged and “removed” from the vector that contains all the tubing-sizes.

After every particle “i” has gone through the system, we will estimate the new absolute permeability and relative injectivity. The number of particles will increase by a hundred thousand and will start at the beginning and run again.

The whole process will stop when we reach 1 million particles. Each the vectors containing the data for absolute permeability and relative injectivity will be plotted. At this point, the MATLAB-script will stop.

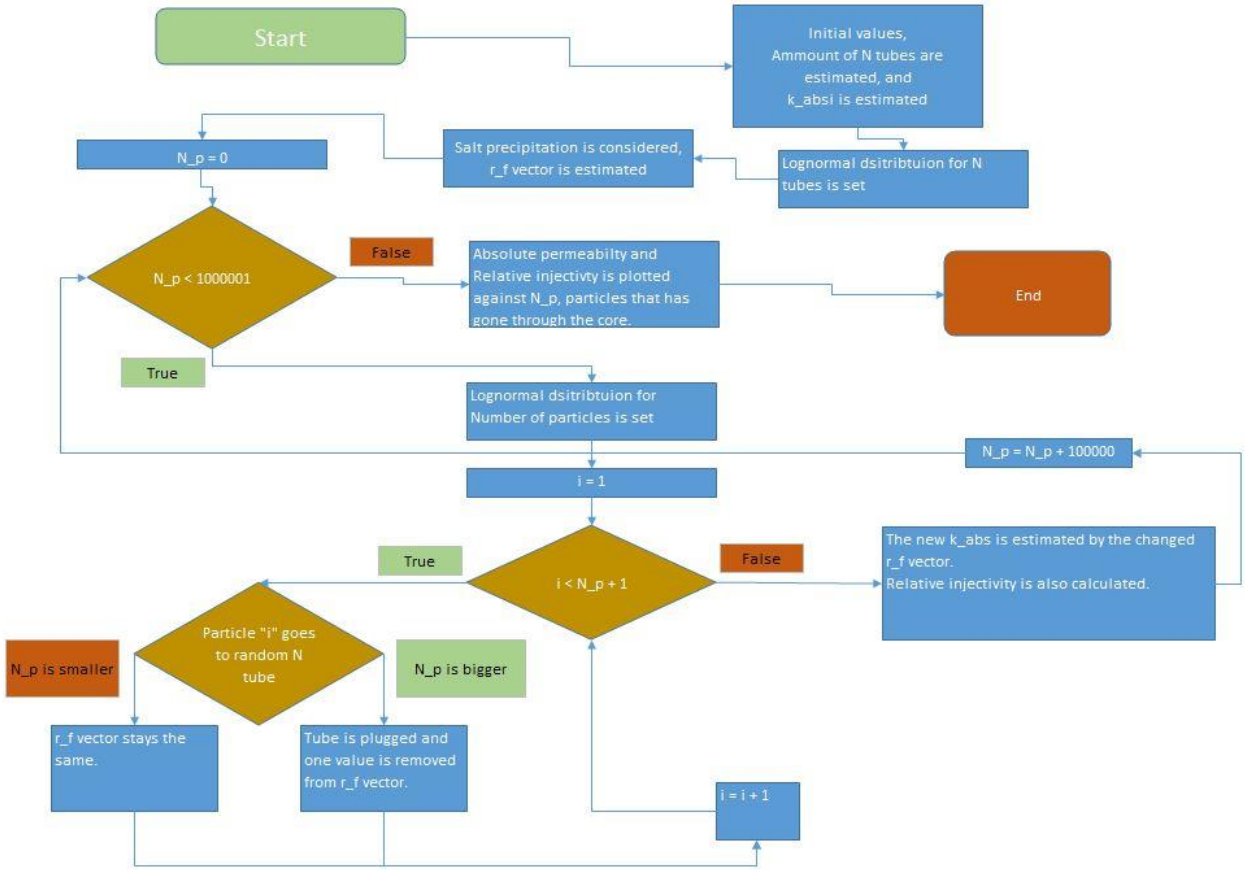

Figure 4-7: A flow chart of the MATLAB file “TurbulentNPart.m”

4.4.2 MATLAB algorithm “LaminarNPart.m”

The algorithm for “LaminarNPart.m” will be the same as “TurbulentNPart.m”, except for some differences:

Two areas are defined for a particle “i” to hit (area A and area B). A particle “i” will have an 80% chance to hit A and a 20% chance to hit B. Particle “i” will then block or go through a tube that lies within the area that it was randomly chosen to go through.

The totally amount of N tubes were estimated in the script firstly. Area A and area B has their own total area, and the N tubes in the entire core were distributed across this area. When for example particle “i” goes through area A and plugs a tube within that area, then area B will remain untouched (and values within B won’t change).

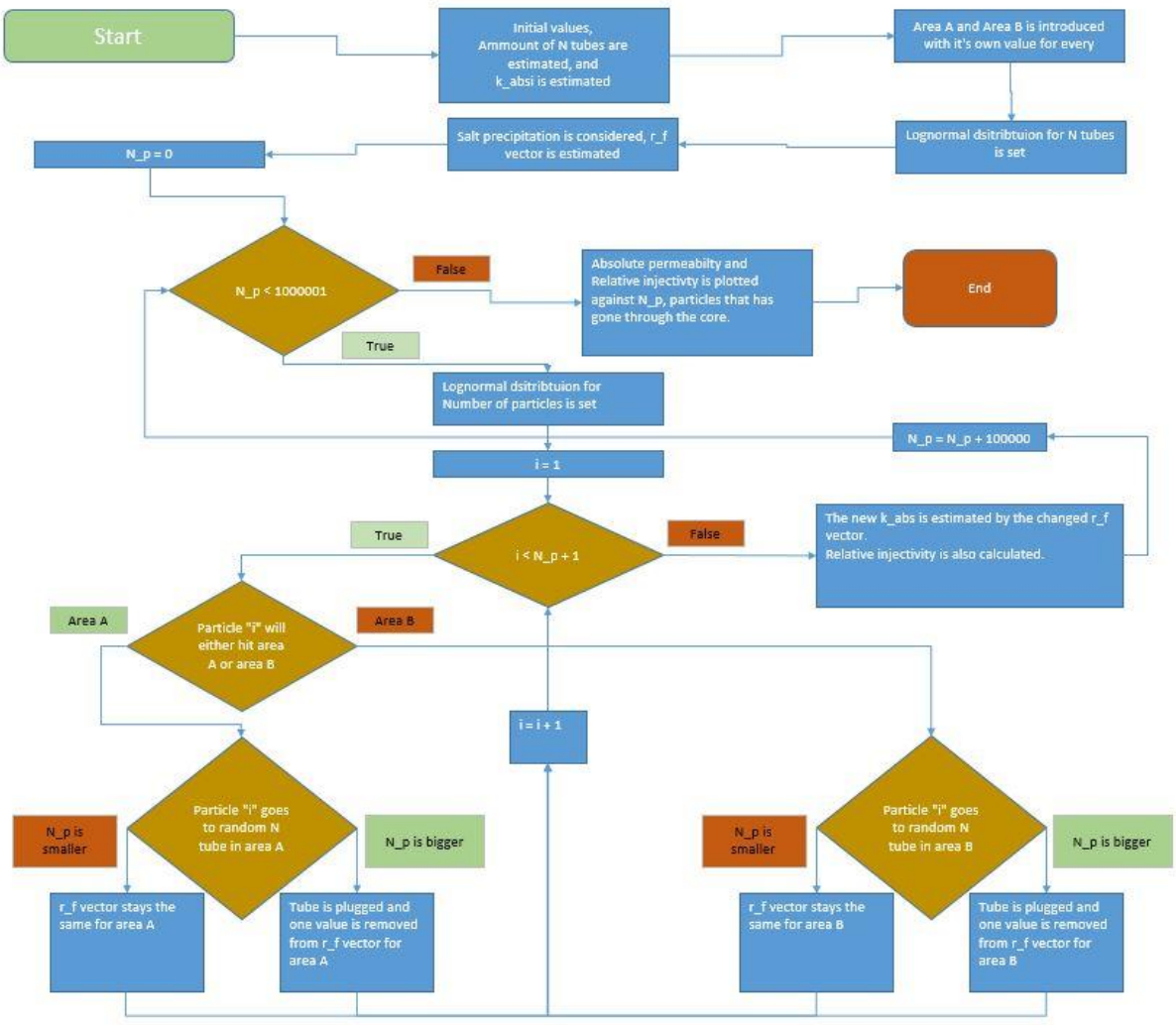

Figure 4-8: A flow chart of the MATLAB file “LaminarNPart.m”

4.5 Inputs for the MATLAB-scripts

Following is a couple of tables that shows us all the inputs that were used in the MATLAB script:

Core and Tubes		
Parameter	Value	Unit
R	2.81/200	m
Average tube radius	6	µm
Statistical mean	1	1
Statistical variance	0,5	1

Table 4-1: Input parameters for the core and tube in the MATLAB script

Particles		
Parameter	Value	Unit
Average particle radius	10	µm
Average tube radius	6	µm
Statistical µ	0	1
Statistical σ	0,5	1

Table 4-2: Input parameters for the particles in the MATLAB script

The MATLAB algorithm was split up into two separate runs, one for a low salt concentration and one for a high salt concentration. Below are tables for input parameters using equation (4.49):

Low salinity			
α	D_{aq}	X_s	D_s
1	1.0974	0.07168	2.16

Table 4-3: Input parameters for a low salt concentration

High salinity			
α	D_{aq}	X_s	D_s
1	1.14875	0.1369	2.16

Table 4-4: Input parameters for a high salt concentration

5 Results and Discussion

5.1 Overview of Results Presentation

The dataset that is generated by the MATLAB-scripts can be summarized in the table below

Flow pattern	Turbulent		Laminar	
Porosity / Salt concentration	0.100	Low	0.100	Low
		High		High
	0.184	Low	0.184	Low
		High		High
	0.300	Low	0.300	Low
		High		High

Table 5-1: Summary of all the dataset generated by the MATLAB-script.

For each dataset that was generated by MATLAB (shown in Table 5-1 above), 1 million particles were ran through the system and the expected particle-sizes was set to 10 micro-meters.

5.1.1 Turbulent flow - Porosity 0.100

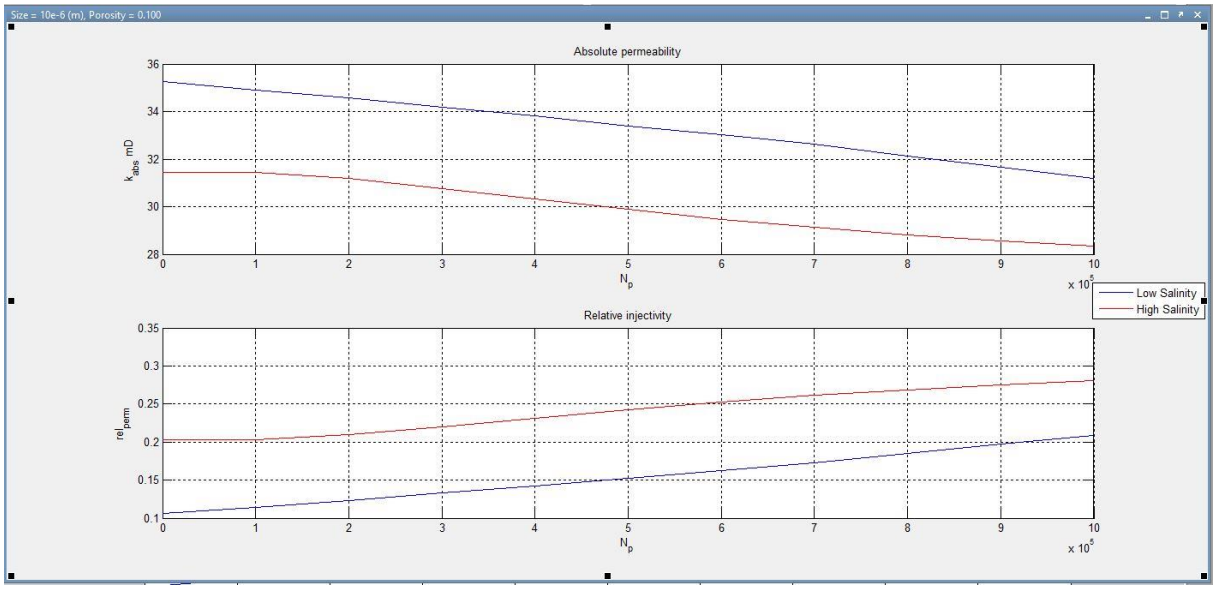

Figure 5-1: Dataset for a turbulent flow, porosity of 0.100

Low salinity		
State	Initial	Final
Abs. perm, mD	35.2524	31.1825
Rel. perm	0.1058	0.2090
High salinity		
State	Initial	Final
Abs. perm, mD	31.4237	28.3436
Rel. perm	0.2029	0.2811

Table 5-2: Exact parameter-numbers as seen in Figure 5-1

From this dataset, we can see that the relative permeability has increased steadily towards a higher value. The impairment due to particles has increased by 0.1032 and 0.0782 for a low salinity and high salinity respectively.

By looking at the relative injectivity in Table 5-4, we can see that the impairment at the initial state is higher for a higher salinity data, than for the lower salinity data. This relative permeability change, by salt, (0.2029) is the same for every case we have a higher salt concentration. The reason behind this is that we have generated a set amount of tubes (equation (4.48)), and blocked them partly by salt (eq. (4.47)). The blockage by salt will always be the same for every tube, even though we may estimate the more tubes by increasing our porosity. The same is true for every experiment where the salinity is low.

5.1.2 Turbulent flow - Porosity 0.184

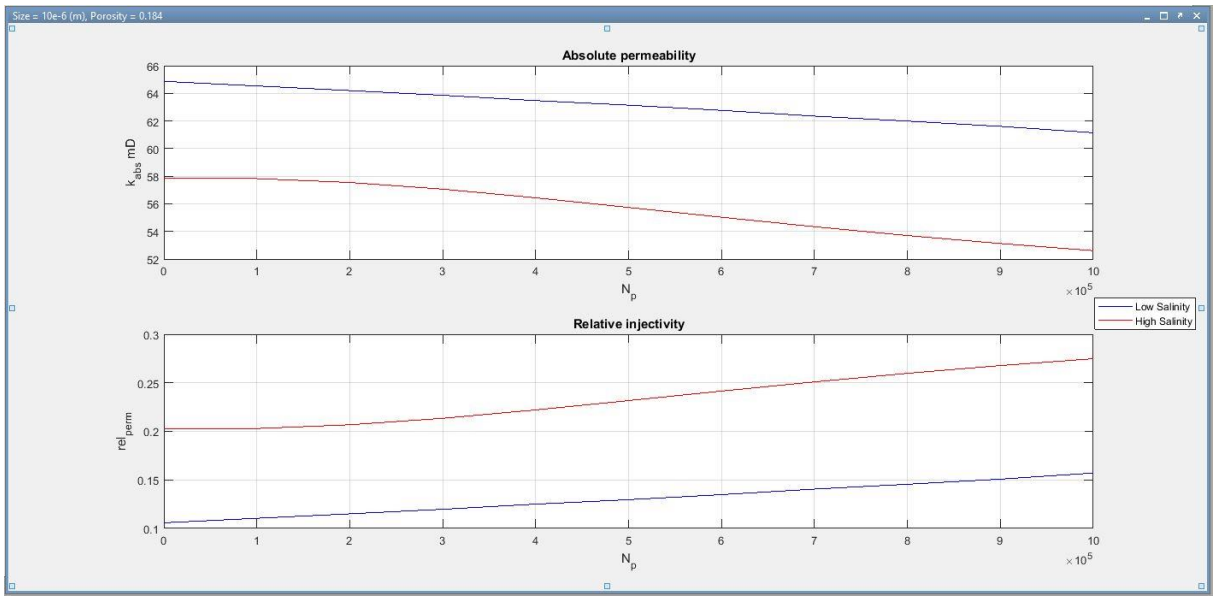

Figure 5-2: Dataset for a turbulent flow, porosity of 0.184

Low salinity		
State	Initial	Final
Abs. perm, mD	64.8695	61.1489
Rel. perm	0.1058	0.1571
High salinity		
State	Initial	Final
Abs. perm, mD	57.8241	52.6050
Rel. perm	0.2029	0.2749

Table 5-3: Exact parameter numbers as seen in Figure 5-2

For this dataset, we can see that particles will reduce the absolute permeabilities as expected. We see that the values for permeability is reduced faster for the "High salinity" curve than for the "Low Salinity" curve. The high salinity state will somewhat increase the amount of tubes that is blocked, compared to the lower salinity state. This is happening, in this model, because the chance that a particle may be captured is higher for the high salinity state, due to the tube sizes being smaller in general because of the higher salt-precipitation blockage. The high salinity state will produce a higher salt salinity, S_s, by equation (4.49). And by equation (4.47) we will generate a higher salt-blockage radius, Δr_i. The high salt concentration is the same as we saw in 5.1.1, but in this case, we have a higher porosity, φ, which generates more tubes in the core. A higher number of tubes combined with the probability density function, which generates both smaller and higher diameter tubes compared to the expected size, will make a higher "availability" for the particles to find tubes to plug.

5.1.3 Turbulent flow - Porosity 0.300

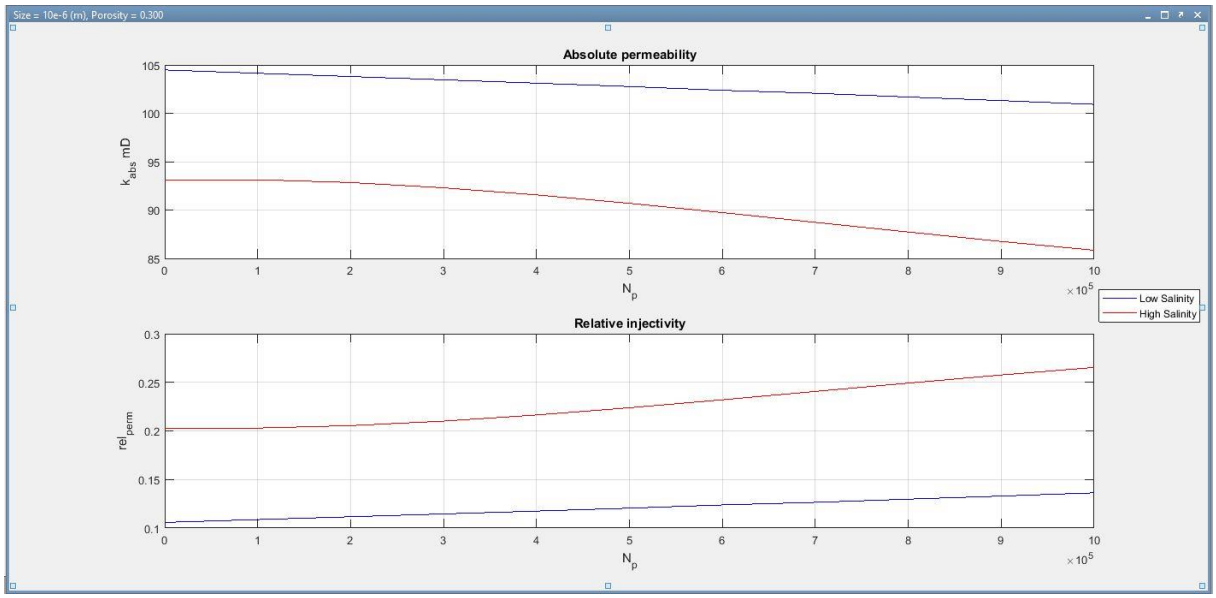

Figure 5-3: Dataset for a turbulent flow, porosity of 0.300

Low salinity		
State	Initial	Final
Abs. perm, mD	104.4821	100.9213
Rel. Inj	0.1058	0.1363
High salinity		
State	Initial	Final
Abs. perm, mD	93.1345	85.8446
Rel. Inj	0.2029	0.2653

Table 5-4: Exact parameter numbers as seen in figure 5-3

For this dataset, we can see that the "high-salinity" values for absolute permeability is decreasing at a more rapid pace when we introduce more particles to the system than with the "low-salinity" values. The porosity is increased compared to 5.1.2, which generates more tubes, N, with equation (4.48). Compared to the datasets in 5.1.1 and 5.1.2, this dataset has a more apparent change for the "high-salinity" values compared to the "low-salinity" values. The impairment in general, which we can measure by the relative injectivity from initial to final, has changed more with the "high-salinity" case. The "high salinity" case gave us a relative injectivity difference of 0.0624, while the "low salinity" case gave us a relative injectivity difference of 0.0305.

If we were to compare the change in relative injectivity for all the turbulent data (5.1.1, 5.1.2 and 5.1.3), we will get as shown in the table below:

Relative permeability changes – impairment by particles			
Porosity	0.100	0.184	0.300
Low salinity	0.1032	0.0513	0.0305
High salinity	0.0782	0.0720	0.0624

Table 5-5: Table comparing relative permeability changes in 5.1.1, 5.1.2 and 5.1.3

Now, lets plot this data seen in Table 5-5. We have porosity on the x-axis, and the relative permeability changes on the y-axis. We make two dataset, where one is representing the "low salinity" values and the other represent the "high salinity" values. Afterwards we create a linear regression through both our dataset, where we also show our regression-number R^2. All of this is done in "Excel". This is shown in the figure below:

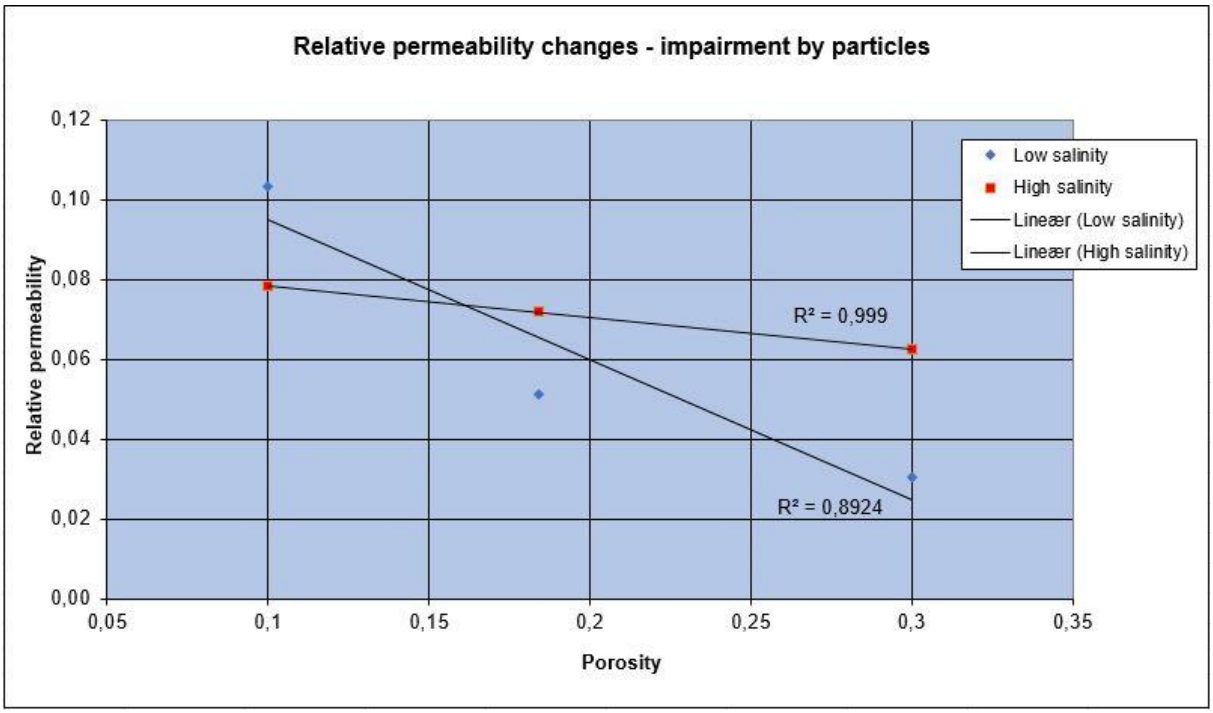

Figure 5-4: Plotting the data seen in Table 5-5

From regression we find that the relative permeability data from our "high salinity" values fit really well. If R^2 is exactly 1, the fit would be perfect. Having a R^2 of 0.999 is very good. But the linear regression for the "low salinity" values are not fitting so well. These values have an R^2 of 0.8924. By looking the points and the linear fit for "low salinity", we can easily spot that the value for a porosity of 0.100 is not fitting so well, compared to the other points. This may suggest that the point (0.100 , 0.1032) is an outlier in this plot.

5.1.4 Laminar flow - Porosity 0.100

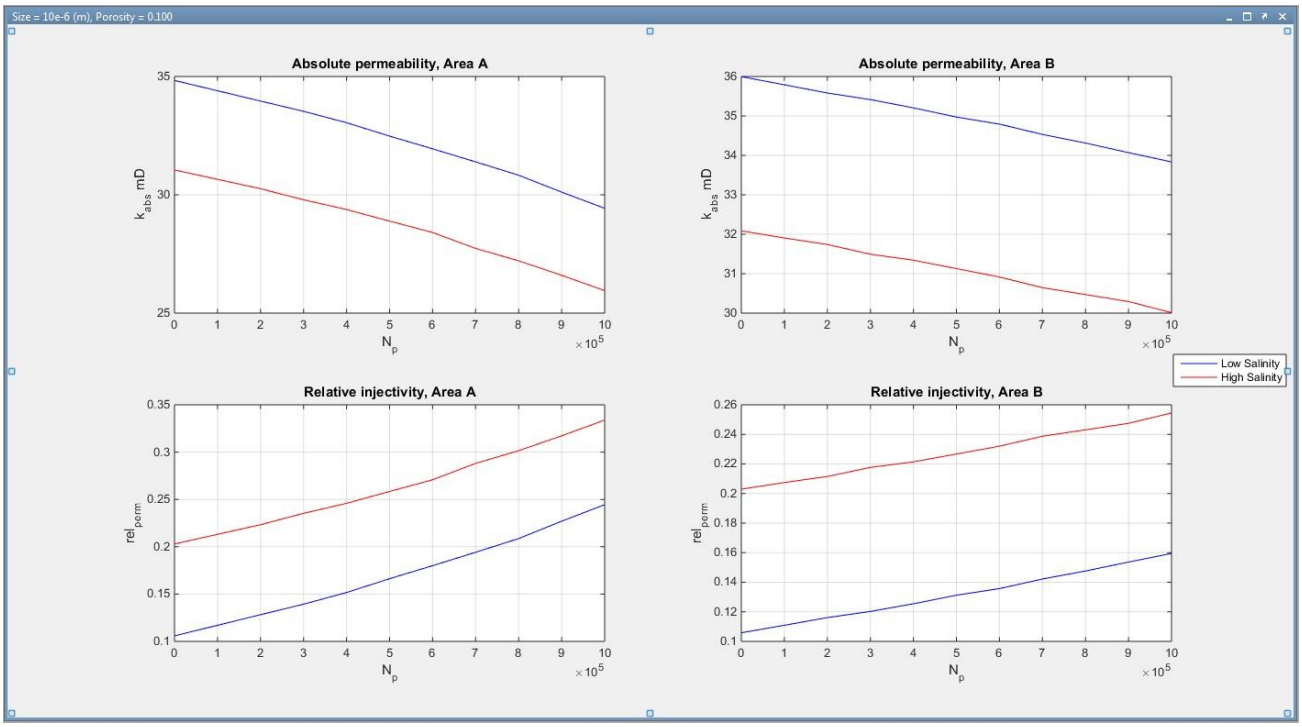

Figure 5-5: Dataset for a laminar flow, porosity of 0.100

Low salinity				
Area	A		B	
State	Initial	Final	Initial	Final
Abs. perm, mD	34.8344	29.4320	35.9956	33.8350
Rel. perm	0.1058	0.2445	0.1058	0.1595
High salinity				
Area	A		B	
State	Initial	Final	Initial	Final
Abs. perm, mD	31.0511	25.9504	32.0862	30.0119
Rel. perm	0.2029	0.3339	0.2029	0.2545

Table 5-6: Exact parameter numbers as seen in figure 5-4

For this dataset, we can see that the impairments are increasing at about the same pace for both the “low salinity”- and “high salinity” values for both area A and B. Since particles were estimated to have a 20 percent chance to enter area B, we can see that, for both salinity sets, that the impairment is significantly less reduces in B than in A.

- Low salinity - Area A: 0.1387
- Low salinity - Area B: 0.0537
- High salinity - Area A: 0.1310
- High salinity - Area B: 0.0516

5.1.5 Laminar flow - Porosity 0.184

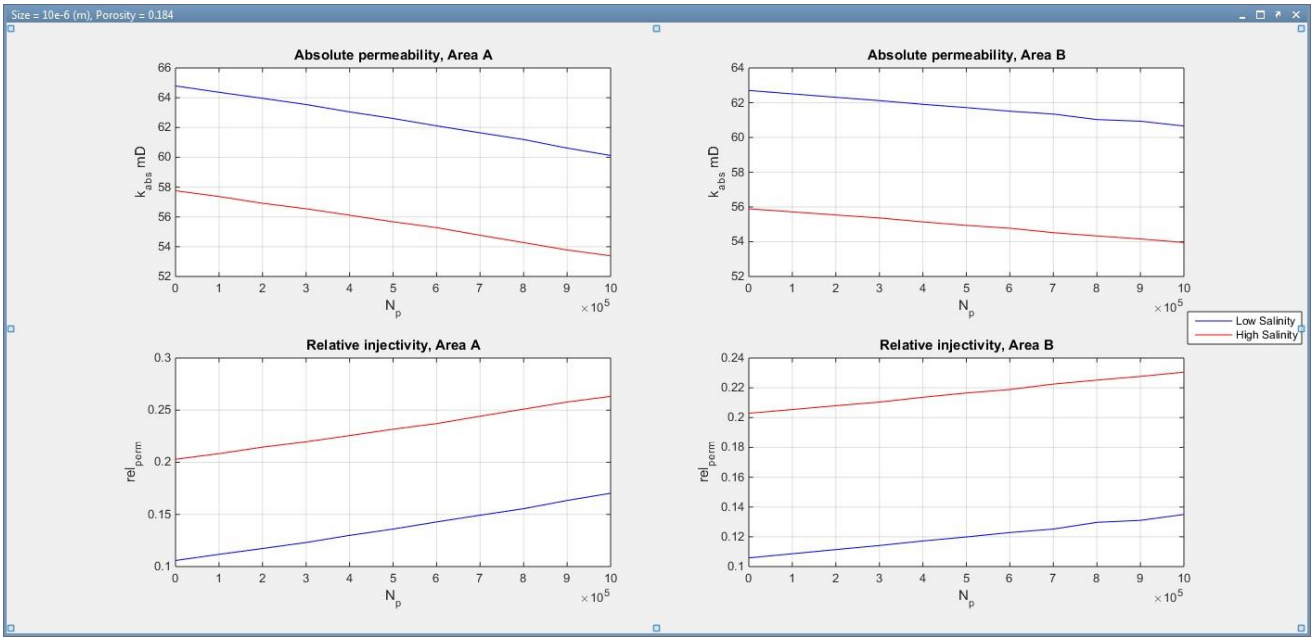

Figure 5-6: Dataset for a laminar flow, porosity of 0.184

Low salinity				
Area	A		B	
State	Initial	Final	Initial	Final
Abs. perm, mD	64.7938	60.1223	62.7033	60.6580
Rel. perm	0.1058	0.1703	0.1058	0.1350
High salinity				
Area	A		B	
State	Initial	Final	Initial	Final
Abs. perm, mD	57.7567	53.3888	55.8932	53.9575
Rel. perm	0.2029	0.2632	0.2029	0.2305

Table 5-7: Exact parameter numbers as seen in Figure 5-5

For this dataset, we can observe somewhat the same pattern we saw in 5.1.4; “Low salinity” values and “High salinity” values for permeabilities decreasing at about the same pace. We can see by Relative permeabilities in Table 5-6 are increasing from the initial to final state, that they increase by about the same; For instance, we can look at area A where the relative permeability change is 0.0645 (0.1703 – 0.1058) for the low salinity case and likewise for the high salinity case it is 0.0603 (0.2632 – 0.2029). The same goes for area B: 0.0292 / 0.0276.

5.1.6 Laminar flow - Porosity 0.300

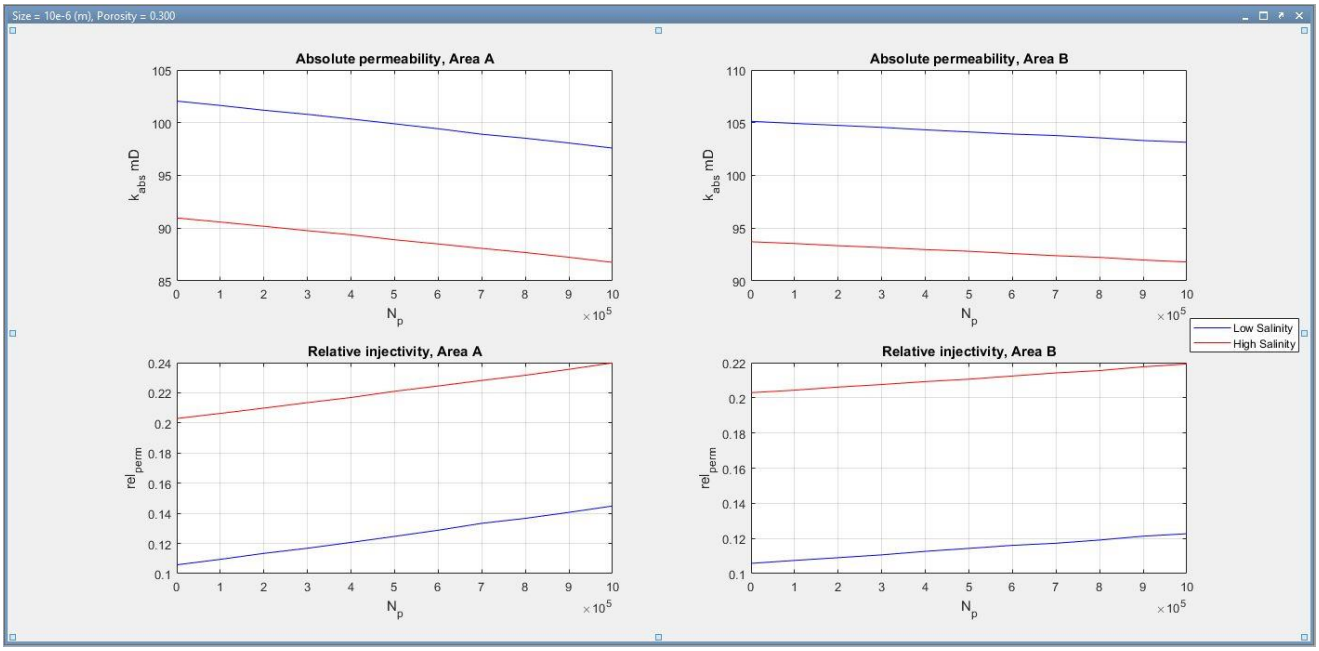

Figure 5-7: Dataset for a laminar flow, porosity of 0.300

Low salinity				
Area	A		B	
State	Initial	Final	Initial	Final
Abs. perm, mD	102.0496	97.5947	105.1284	103.1456
Rel. perm	0.1058	0.1448	0.1058	0.1227
High salinity				
Area	A		B	
State	Initial	Final	Initial	Final
Abs. perm, mD	90.9662	86.7544	93.7106	91.7905
Rel. perm	0.2029	0.2398	0.2029	0.2193

Table 5-8: Exact parameter numbers as seen in figure 5-6

The same trend that was discussed in 5.1.4 and 5.1.5 is valid for this dataset as well. By looking at the relative permeabilities again we can see how the impairment in the core is building up as the particles run through the system:

- Low salinity - Area A: 0.0390
- Low salinity - Area B: 0.0169
- High salinity - Area A: 0.0369
- High salinity - Area B: 0.0164

5.2 Pore-Size Distribution

Using the probability density function to generate a distribution of tubing sizes will look like the one shown in Figure 5-7. The expected tubing size were set to 6µm, which was the same for each iteration run in the MATLAB script. Figure 5-7 was generated by putting the porosity, ϕ, equal to 0.184. The statistical parameters were as follows: statistical mean (m) = 1, statistical variance (v) = 0.5. Running these parameters through eqution (4.52) and (4.53), will generate the PDF seen below by equation (4.54).

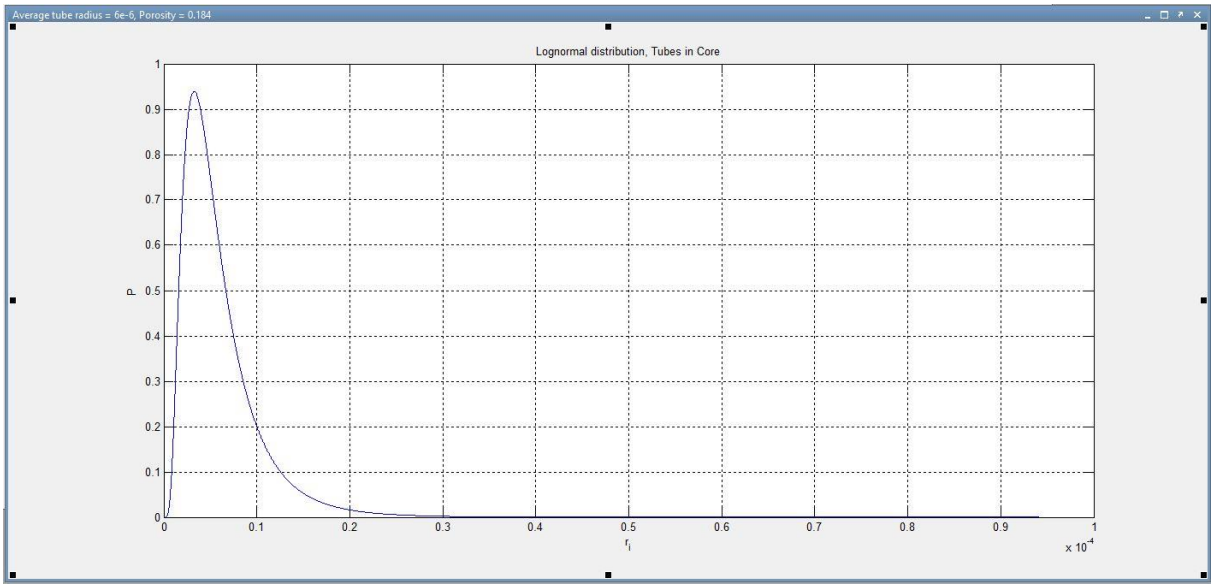

Figure 5-8: PDF function showing how tubes are distributed in the model

5.3 Effect of Particle Size

In this section I will look at the differences for some datasets, where the particle size has been increased from 10 micro-meters to 15 micro-meters. For both datasets, we use a porosity, ϕ, of 0.184, and put it within turbulent flow regime.

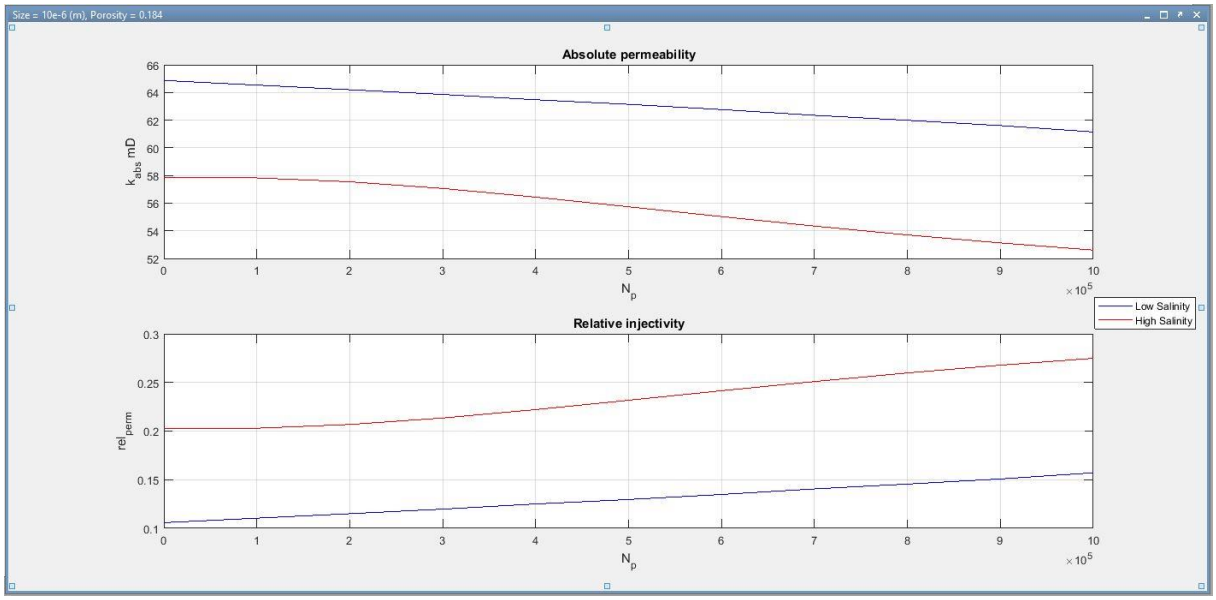

Figure 5-9: Same dataset as seen in figure 5-5

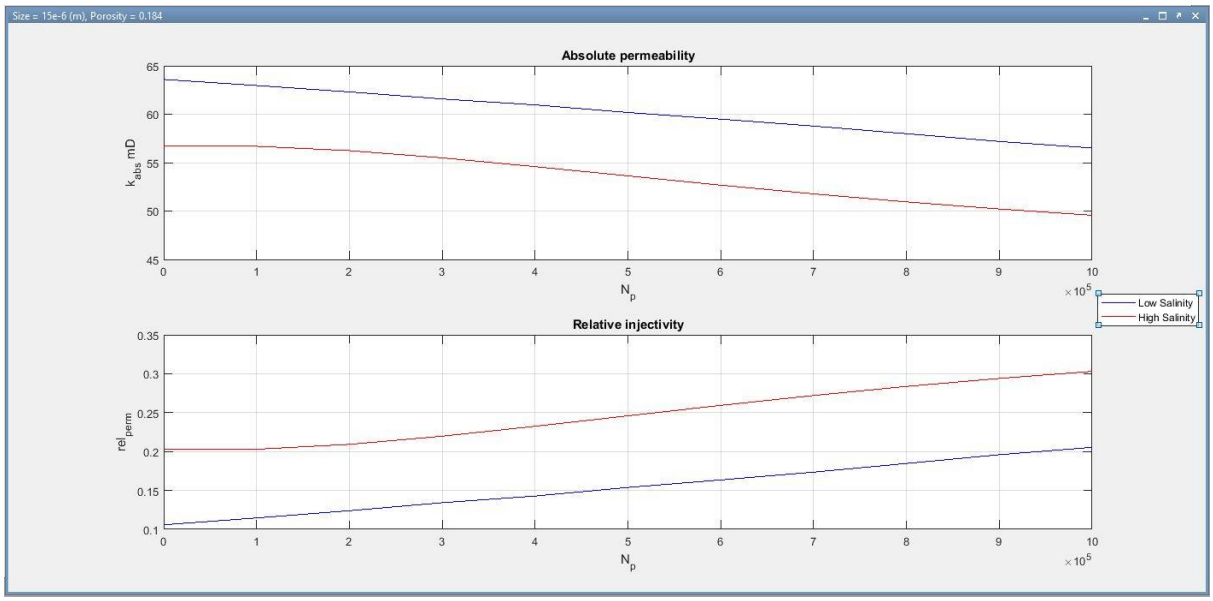

Figure 5-10: Dataset for turbulent flow, porosity of 0.184 and particle size 15 µm

Low Salinity				
Particle size, µm	10		15	
State	Initial	Final	Initial	Final
Abs. perm, mD	64.8695	61.1489	63.5939	56.5135
Rel. perm	0.1058	0.1571	0.1058	0.2054
High Salinity				
Particle size, µm	10		15	
State	Initial	Final	Initial	Final
Abs. perm, mD	57.8241	52.6050	56.6871	49.5673
Rel. perm	0.2029	0.2749	0.2029	0.3030

Table 5-9: A comparison of the dataset seen in figure 5-7 and figure 5-8

For this dataset one thing is immediately apparent: A higher particle size, will increase the amount of impairment that happens during colloidal particles against the tubes in the core. We can see that the relative permeability for both "low salinity" and "high salinity" starts at the same value, since particles hasn't had a chance to flow against the core.

The permeability impairment, caused by colloidal particles, has a bigger effect when the salt is less of a problem, ergo in the "low salinity" case. The Relative permeability at the final state is 0.1571 for the particles at 10 µm size. In the case for a particle size of 15 µm, then the relative permeability will be 0.2054.

If we look at the case where we have a higher "salt salinity", we will find that the particle sizing has less of an effect. This may be due to the tubing are already been reduced to such a size by the salt, that increasing the particle size won't matter. To elaborate, this means that in some cases a 10 µm will block most of the tubes, and if a 15 µm particle come it won't matter. Below are all the relative permeability changes find in Table 5-9:

- $R_{particle}$, µm = 10 – low salinity: 0.0513
- $R_{particle}$, µm = 15 – low salinity: 0.0996
- $R_{particle}$, µm = 10 – high salinity: 0.0720
- $R_{particle}$, µm = 15 – high salinity: 0.1001

5.4 Effect of Initial Permeability

The initial permeabilities without any salt precipitation was estimated by equation 4.50. Without any salt precipitation Δr^2 will be equal to zero. The only thing that will change the permeability with any salt is the porosity, ϕ, in our case. Increasing the porosity will increase the number of tubes, estimated by equation (4.48). Decreasing the porosity will likewise, decrease the number of tubes. By having more/less tubes (N), we will increase/decrease the number of terms in the sum in equation (4.50). Our non-salt permeabilities can be presented as such:

No salt precipitation / Turbulent flow			
ϕ	0,1	0,184	0,3
Abs. perm, mD	39.4241	72.5459	116.8461

Table 5-10: Absolute permeabilizes with no salt-precipitation in a turbulent flow regime.

No salt precipitation / Laminar flow						
ϕ	0,1		0,184		0,3	
Area	A	B	A	B	A	B
Abs. perm, mD	38.9566	40.2552	72.4612	70.1233	114.1258	117.5688

Table 5-11: Absolute permeabilities with no-salt precipitation in a laminar flow regime.

As stated above we can see that porosity, Φ, will increase the absolute permeabilities. These numbers will act as an estimation of how the permeabilities in the core act.

Below is a table to represent the effect of the permeability after the initial permeabilities:

Φ = 0.184								
Low salinity					High salinity			
Area	A		B		A		B	
State	Initial	Final	Initial	Final	Initial	Final	Initial	Final
Abs. perm, mD	64.7938	60.1223	62.7033	60.6580	57.7567	53.3888	55.8932	53.9575
Rel. perm	0.1058	0.1703	0.1058	0.1350	0.2029	0.2632	0.2029	0.2305
Φ = 0.300								
Low salinity					High salinity			
Area	A		B		A		B	
State	Initial	Final	Initial	Final	Initial	Final	Initial	Final
Abs. perm, mD	102.0496	97.5947	105.1284	103.1456	90.9662	86.7544	93.7106	91.7905
Rel. perm	0.1058	0.1448	0.1058	0.1227	0.2029	0.2398	0.2029	0.2193

Table 5-12: Table is presenting the differences in two datasets at different porosities.

The dataset for Laminar flow - Porosity 0.184 and Laminar flow - Porosity 0.300 is used to get the dataset for the table above.

The initial permeability that is changed by the porosity will change things up. We can see that the relative permeability changes, from initial to final state, will be much less when porosity rises to a higher level; here: 0.300. The relative permeabilities changes as follows:

- Φ = 0.184 – low salinity – area A: 0.0645
- Φ = 0.184 – low salinity – area B: 0.0292
- Φ = 0.184 – high salinity – area A: 0.0603
- Φ = 0.184 – high salinity – area B: 0.0276
- Φ = 0.300 – low salinity – area A: 0.0390
- Φ = 0.300 – low salinity – area B: 0.0169
- Φ = 0.300 – high salinity – area A: 0.0369
- Φ = 0.300 – high salinity – area B: 0.0164

Having a higher porosity will give more amount of tubes, N, by equation (4.48) and therefore giving a higher permeability, k_{abs}, by equation (4.50). This means there are more tubes to plug in the core, making it require more particles to make an equally high impairment effect as in the lower porosity case.

6 Conclusion

6.1 Summary and Highlights

As we saw with the turbulent data (5.1.1, 5.1.2, 5.1.3), we may have detected an increased difference, for relative permeability, in the “high salinity” values compared to the “low salinity” values. The relative seemed to increase abit faster as the porosity of the dataset increased gradually ([0.100, 0.184, 0.300]). For a turbulent flow with a porosity of 0.100, we got a bigger difference in relative permeability for “low salinity” than for “high salinity”. This was abit unexpected, because it makes more sense for particles to reduce permeability in a “higher salt concentration”-case. In section 5.1.3, we looked at the possibility of the relative permeability change at 0.100 porosity being an outlier in a trend. The MATLAB-script is setup so that our particle “i” will roll randomly to find a tubing to plug (if it’s bigger) or go through (if it’s smaller). This creates the possibility of sometimes going against the odds. If this point was meant to be lower, then the increased difference in relative permeability when increasing the porosity value would make more sense.

In section 5.3, we looked at the effect of increased particle sizes. We saw that increased particles would indeed have a reducing effect on the permeability. Increasing the salt concentration (going from low salinity to high salinity) also has a reducing effect on permeability. However, in our data where we look at the relative permeability changes going from “low salinity” to “high salinity” compared to going from 10 micro-meter to 15 micro-meter particles, we see some differences. Going from “low salinity” – 10 micro-meter particles to “low salinity” – 15 micro-meter particles, yields a relative permeability difference of 0.0483. If we go from “low salinity” – 10 micro-meter particles to “high salinity” – 15 micro-meter particles, we will get the relative permeability difference of 0.0207. In these data it clearly shows that increasing particle sizes has a much bigger effect on how the relative permeability increases.

In section 5.4, we looked at the effect of initial permeabilities. The only way we can change this number, according to our model, is by changing the porosity. We changed the porosity from 0.184 to 0.300. We used the data from 5.1.5 and 5.1.6, which is laminar data for a porosity of 0.184 and 0.300 respectively. It was clearly shown, by the data, that if we were to increase the porosity, the impairment caused by particles would also decrease. The reason behind this, is that the particles has many more tubes to plug. This means, that we would have

to send many more particles through the core, to reduce the permeability at a higher porosity compared to a lower one. If we compare the relative permeability chamges from out data, where we take the difference from the porosity change (0.184 to 0.300), we will get:

- Low salinity – area A: 0.0645-0.0603 = 0.0255
- Low salinity – area B: 0.0123
- High salinity – area A: 0.0234
- High salinity – area B: 0.0112

There is about 8-9% change in the "relative permeability changes" when going from a porosity of 0.184 to 0.300.

6.2 Proposed Further Work

For further work I would propose that more parameters could be put into a mathematical program such as MATLAB. A concentration of particles could be integrated, so that we could measure how such a concentration would bombard the core with particles over time and cause reduced permeabilities. Having certain areas of the core surface starting to fill up with particles, due to high activity at some spots of the core, would also be quite interesting. Trying to compare more particle sizes, would be resourceful. This would give a clearer picture of how salinity changes versus particle sizes would act.

More dataset will always be good to have. With more dataset, outliers should be easier to spot, and trends should be easier to spot. For reference, each run by my MATLAB-script took 8 to 12 hours to complete. Adding more things for the script to run through, would add to that time.

7 References

Andre, L., Peysson, Y., & Azaroual, M. (2014). Well injectivity during CO 2 storage operations in deep saline aquifers–Part 2: Numerical simulations of drying, salt deposit mechanisms and role of capillary forces. *International Journal of Greenhouse Gas Control, 22*, 301-312.

Avila, K., Moxey, D., Lozar, A. d., Avila, M., Barkley, D., & Hof, B. (2011). *The Onset of Turbulence in Pipe Flow*.

Batchelor, G. K. (2000). *An introduction to fluid dynamics*: Cambridge university press.

Birkholzer, J. T., Oldenburg, C. M., & Zhou, Q. (2015). CO 2 migration and pressure evolution in deep saline aquifers. *International Journal of Greenhouse Gas Control, 40*, 203-220.

Cath, N., & Andrew, S. (2009). Real Fluids - An Introduction to Fluid Mechanics. Retrieved from http://www.efm.leeds.ac.uk/CIVE/CIVE1400/Section4/laminar_turbulent.htm

CCSA. (2011-2017a). Frequently asked questions - CCS general. Retrieved from http://www.ccsassociation.org/faqs/ccs-general/

CCSA. (2011-2017b). Tackling Climate Change. Retrieved from http://www.ccsassociation.org/why-ccs/tackling-climate-change/

CCSA. (2011-2017c). Why CCS? Retrieved from http://www.ccsassociation.org/why-ccs/

Darcy, H. (1856). *Les fontaines publiques de la ville de Dijon: exposition et application*: Victor Dalmont.

Dullien, F. A. (2012). *Porous media: fluid transport and pore structure*: Academic press.

Dvorkin, J. (2009). Kozeny-Carman equation revisited. *unpublished paper, URL< http://pangea.stanford.edu/~jack/KC_2009_JD.pdf >[February 2013]*.

Gibbins, J., & Chalmers, H. (2008). Carbon capture and storage. *Energy policy, 36*(12), 4317-4322.

Giorgis, T., Carpita, M., & Battistelli, A. (2007). 2D modeling of salt precipitation during the injection of dry CO 2 in a depleted gas reservoir. *Energy Conversion and Management, 48*(6), 1816-1826.

Grude, S., Landrø, M., & Dvorkin, J. (2014). Pressure effects caused by CO 2 injection in the Tubåen Fm., the Snøhvit field. *International Journal of Greenhouse Gas Control, 27*, 178-187.

Ives, K. (1985). Deep bed filters *Mathematical models and design methods in solid-liquid separation* (pp. 90-332): Springer.

Khilar, K., & Fogler, H. (1998). *Migration of Fines in Porous Media. Theory and Applications of Transport in Porous Media, vol. 12*: Kluwer Academic Publishing, Dordrecht.

Kim, K.-Y., Han, W. S., Oh, J., Kim, T., & Kim, J.-C. (2012). Characteristics of salt-precipitation and the associated pressure build-up during CO2 storage in saline aquifers. *Transport in Porous Media, 92*(2), 397-418.

Lin, C., & Cohen, M. (1982). Quantitative methods for microgeometric modeling. *Journal of Applied Physics, 53*(6), 4152-4165.

Lymberopoulos, D., & Payatakes, A. (1992). Derivation of topological, geometrical, and correlational properties of porous media from pore-chart analysis of serial section data. *Journal of colloid and interface science, 150*(1), 61-80.

McDowell-Boyer, L. M., Hunt, J. R., & Sitar, N. (1986). Particle transport through porous media. *Water Resources Research, 22*(13), 1901-1921.

Miri, R., & Hellevang, H. (2016). Salt precipitation during CO 2 storage—A review. *International Journal of Greenhouse Gas Control, 51*, 136-147.

Miri, R., van Noort, R., Aagaard, P., & Hellevang, H. (2015). New insights on the physics of salt precipitation during injection of CO 2 into saline aquifers. *International Journal of Greenhouse Gas Control, 43*, 10-21.

Muller, N., Qi, R., Mackie, E., Pruess, K., & Blunt, M. J. (2009). CO2 injection impairment due to halite precipitation. *Energy procedia, 1*(1), 3507-3514.

Nave, R. (2005). HyperPhysics. Retrieved from http://hyperphysics.phy-astr.gsu.edu/hbase/pfric.html

Norton, J. D. (2005). Atoms and the Quantum. Retrieved from http://www.pitt.edu/~jdnorton/teaching/HPS_0410/chapters_2017_Jan_1/atoms_quantum/index.html#L4406

Ott, H., De Kloe, K., Marcelis, F., & Makurat, A. (2011). Injection of supercritical CO2 in brine saturated sandstone: Pattern formation during salt precipitation. *Energy procedia, 4*, 4425-4432.

Ott, H., Roels, S., & De Kloe, K. (2015). Salt precipitation due to supercritical gas injection: I. Capillary-driven flow in unimodal sandstone. *International Journal of Greenhouse Gas Control, 43*, 247-255.

Ott, H., Snippe, J., De Kloe, K., Husain, H., & Abri, A. (2013). Salt precipitation due to Sc-gas injection: single versus multi-porosity rocks. *Energy procedia, 37*, 3319-3330.

Peysson, Y. (2012). Permeability alteration induced by drying of brines in porous media. *The European Physical Journal Applied Physics, 60*(2), 24206.

Peysson, Y., Andre, L., & Azaroual, M. (2014). Well injectivity during CO 2 storage operations in deep saline aquifers—Part 1: Experimental investigation of drying effects, salt precipitation and capillary forces. *International Journal of Greenhouse Gas Control, 22*, 291-300.

Pruess, K. (2009). Formation dry-out from CO2 injection into saline aquifers: 2. Analytical model for salt precipitation. *Water Resources Research, 45*(3).

Pruess, K., & Müller, N. (2009). Formation dry-out from CO2 injection into saline aquifers: 1. Effects of solids precipitation and their mitigation. *Water Resources Research, 45*(3).

Richard, F. (1970). *The Feynman Lectures on Physics Vol I*.

Roels, S. M., Ott, H., & Zitha, P. L. (2014). μ-CT analysis and numerical simulation of drying effects of CO 2 injection into brine-saturated porous media. *International Journal of Greenhouse Gas Control, 27*, 146-154.

Scheidegger, A. (1974). The physics of flow through porous media. University of Toronto: Toronto.

Schembre-McCabe, J. M., Kamath, J., & Gurton, R. M. (2007). *Mechanistic studies of CO2 sequestration.* Paper presented at the International Petroleum Technology Conference.

Shahidzadeh-Bonn, N., Rafaï, S., Bonn, D., & Wegdam, G. (2008). Salt crystallization during evaporation: impact of interfacial properties. *Langmuir, 24*(16), 8599-8605.

Solomon, S., Plattner, G.-K., Knutti, R., & Friedlingstein, P. (2009). Irreversible climate change due to carbon dioxide emissions. *Proceedings of the national academy of sciences*, pnas. 0812721106.

Wikipedia. (2017). Brownian motion. Retrieved from https://en.wikipedia.org/wiki/Brownian_motion

Wojtanowicz, A., Krilov, Z., & Langlinais, J. (1987). *Study on the effect of pore blocking mechanisms on formation damage.* Paper presented at the SPE Production Operations Symposium.

Wojtanowicz, K. A., & Krilov, J. P. L. Z. (1988). Experimental Determination of Formation Damage Pore Blocking Mechanisms. *Journal of Energy Resources Technology*, 34-42.

Zeidouni, M., Pooladi-Darvish, M., & Keith, D. (2009). Analytical solution to evaluate salt precipitation during CO 2 injection in saline aquifers. *International Journal of Greenhouse Gas Control, 3*(5), 600-611.

8 Appendix

8.1 Appendix A: MATLAB-script “TurbulentNPart.m”

```
%% Run 1 - low salt concentration
clc
clear
clf

% Conversion factor
Conv_mD = 1.01325.*10e12;                                   % Multiply SI unit with this to get mD (mD/m^2)

% Particles info
dens_p = 1.26;                                              % Fumed alumina density (g/cc)
part_rad_avg = 10e-6;                                       % Expected size of particles in fluid

% Estimate N, the average number of tubings in the core
poro_init = 0.184;                                          % Initial porosity
R = 3.81/200;                                               % Core radius
tube_rad_avg = 6e-6;                                        % Average tube radius
N =int32(3/4.*poro_init.*(R/tube_rad_avg).^2);

% Lognormal distribution of tubes in the core
m = 1;                                                      % mean value
v = 0.5;                                                    % variance
mu = log((m^2)/sqrt(v+m^2));                                % myu for lognormal function
sigma = sqrt(log(v/(m^2)+1));                               % sigma for lognormal function

rad_tube = lognrnd(mu,sigma,[1,N]);                         % random number generated from lognormal distribution
rad_tube = sort(rad_tube);                                  % sort the random number from small to large
P = lognpdf(rad_tube,mu,sigma);                             % create distribution function from random number generated
rad_tube1 = rad_tube*tube_rad_avg;                          % convert the random number to micrometer size

% Lognormal distribution for the particles has to be moved to the loop...

% Reynolds number
Q = 0.01;                                                   % (m^3/s)
D_core = R*2;                                               % (m)
dyn_visc = 50/1000;                                         % Dynamic viscosity (Pa*s)
kin_visc = dyn_visc./(dens_p*1000);                         % kinematic viscosity (m^2/s)
kin_visc_cSt = kin_visc*10e6;                               % kinematic viscosity centi-stokes (cSt)
A_core = pi()*R.^2;                                         % Area of core (m)
Re = (Q*D_core) / (kin_visc*A_core);                        % Reynolds number


% Absolute permeability without salt precipitation:
k_absi_nosalt = (1./(8*(R.^2))*sum(rad_tube1.^4));
k_absi_nosalt_mD = k_absi_nosalt * Conv_mD;

% Estimating delta r, the salt thickness
dry_coef = 1;
D_aq = 1.0974;
X_s = 0.07168;
D_aq1 = 1.14875;
X_s1 = 0.1369;
D_s = 2.16;

salt_sat = (0.85 + dry_coef/3.5)*(D_aq*X_s/D_s);            % solid salt saturation

del_r = 2/3*(rad_tube1*salt_sat)/dry_coef;                  % delta r
r_f = rad_tube1 - del_r;                                    % r - delta_r


% Calculation of absolute permeability k_abs initial
k_absi = (1./(8*(R.^2)))*sum(r_f.^4);
k_absi_mD = k_absi * Conv_mD;

% Vectors to be plotted
k_abs_vector = [];
k_abs_vector(1,1) = k_absi_mD;

rel_perm_vector = [];
rel_perm_vector(1,1) = 1 - (k_absi_mD / k_absi_nosalt_mD);

N_p_vector = [];
N_p_vector(1,1) = 0;


N_p = 0;                                                    % Number of particles in a cubic meter
N_p_step = 100000;                                          % The ammount of particles in each step
j = 1;                                                      % Step counter for plotting the vectors
r_count = 0;                                                % Reset counter for when we reduce the tubes N

while N_p < 1000001
r_f_redu = r_f;

% Lognormal distribution of particles in the core
%m = 0;                                                     % mean value
%v = 0.25;                                                  % variance
mu_part = 0;                                                % myu for lognormal function
sigma_part = 0.5;                                           % sigma for lognormal function

part_dist = lognrnd(mu_part,sigma_part,[1,N_p]);            % random number generated from lognormal distribution
part_dist = sort(part_dist);                                % sort the random number from small to large
P_part = lognpdf(part_dist,mu,sigma);                       % create distribution function from random number generated
part_dist1 = part_dist*part_rad_avg;                        % convert the random number to micrometer size


    i = 1;                                                  % Particle "i" that will act upon the system of tubes
    while i < N_p + 1                                       % Run every particle
```

```
        if N>0 && N_p>0
            elim = randi(N);
            elimp = randi(N_p);

            if part_dist1(1,elimp) < r_f_redu(1,elim)
            else
            part_dist1(1,elimp) = 0;
            part_dist1 = part_dist1(0~=part_dist1);
            N_p = N_p - 1;

            r_f_redu(1,elim) = 0;                          % Eliminate tube where the size of the particle is greater
            r_f_redu = r_f_redu(0~=r_f_redu);              % Eliminating zeros
            N = N - 1;                                     % Reduce the N ammount of tubes when plugged
            r_count = r_count + 1;
            end
        else
        end
    i = i + 1;
    end

% Reseting N tubes
N = N + r_count;
N_p = N_p + r_count;
r_count = 0;

% Getting values for the vectors
k_abs = (1./(8*(R.^2)))*sum(r_f_redu.^4);

k_abs_mD = k_abs * Conv_mD;
rel_perm = 1 - (k_abs_mD / k_absi_nosalt_mD);

k_abs_vector(1,j+1) = k_abs_mD;
rel_perm_vector(1,j+1) = rel_perm;
N_p_vector(1,j+1) = N_p;

N_p = N_p + N_p_step
j = j + 1;
end

% Run 2 - high salt concentration

% Not reapeating constants

% Estimating delta r, the salt thickness
dry_coef = 1;
D_aq = 1.0974;
X_s = 0.07168;
D_aq1 = 1.14875;
X_s1 = 0.1369;
D_s = 2.16;

salt_sat1 = (0.85 + dry_coef/3.5)*(D_aq1*X_s1/D_s);         % solid salt saturation

del_r1 = 2/3*(rad_tube1*salt_sat1)/dry_coef;                % delta r
r_f1 = rad_tube1 - del_r1;                                  % r - delta_r


% Calculation of absolute permeability k_abs initial
k_absi1 = (1./(8*(R.^2)))*sum(r_f1.^4);
k_absi_mD1 = k_absi1 * Conv_mD;

% Vectors to be plotted
k_abs_vector1 = [];
k_abs_vector1(1,1) = k_absi_mD1;

rel_perm_vector1 = [];
rel_perm_vector1(1,1) = 1 - (k_absi_mD1 / k_absi_nosalt_mD);

N_p_vector1 = [];
N_p_vector1(1,1) = 0;

N_p1 = 0;                                                   % Number of particles in a cubic meter
N_p_step1 = 100000;                                         % The ammount of particles in each step
j = 1;                                                      % Step counter for plotting the vectors
r_count = 0;                                                % Reset counter for when we reduce the tubes N

while N_p1 < 1000001
r_f_redu1 = r_f1;

% Lognormal distribution of particles in the core
%m = 0;                                                     % mean value
%v = 0.25;                                                  % variance
mu_part = 0;                                                % myu for lognormal function
sigma_part = 0.5;                                           % sigma for lognormal function

part_dist = lognrnd(mu_part,sigma_part,[1,N_p1]);           % random number generated from lognormal distribution
part_dist = sort(part_dist);                                % sort the random number from small to large
P_part = lognpdf(part_dist,mu,sigma);                       % create distribution function from random number generated
part_dist1 = part_dist*part_rad_avg;                        % convert the random number to micrometer size


    i = 1;                                                  % Particle "i" that will act upon the system of tubes
    while i < N_p1 + 1                                      % Run every particle

        if N>0 && N_p1>0
            elim = randi(N);
            elimp = randi(N_p1);

            if part_dist1(1,elimp) < r_f1(1,elim)
            else
            part_dist1(1,elimp) = 0;
            part_dist1 = part_dist1(0~=part_dist1);
            N_p1 = N_p1 - 1;
```

```
            r_f1(1,elim) = 0;                                   % Eliminate tube where the size of the particle is greater
            r_f_redu1 = r_f_redu1(0~=r_f_redu1);                % Eliminating zeros
            N = N - 1;                                          % Reduce the N ammount of tubes when plugged
            r_count = r_count + 1;
            end
        else
        end

    i = i + 1;
    end

% Reseting N tubes
N = N + r_count;
N_p1 = N_p1 + r_count;
r_count = 0;

% Getting values for the vectors
k_abs1 = (1./(8*(R.^2)))*sum(r_f_redu1.^4);

k_abs_mD1 = k_abs1 * Conv_mD;
rel_perm1 = 1 - (k_abs_mD1 / k_absi_nosalt_mD);

k_abs_vector1(1,j+1) = k_abs_mD1;
rel_perm_vector1(1,j+1) = rel_perm1;
N_p_vector1(1,j+1) = N_p1;

N_p1 = N_p1 + N_p_step1
j = j + 1;
end


%% Plotting

figure('Name','Size = 10e-6 (m), Porosity = 0.184')
subplot(2,1,1)
plot(N_p_vector,k_abs_vector,'b',N_p_vector1,k_abs_vector1,'r')
grid on
title('Absolute permeability')
xlabel('N_p')
ylabel('k_a_b_s mD')

subplot(2,1,2)
plot(N_p_vector,rel_perm_vector,'b',N_p_vector1,rel_perm_vector1,'r')
grid on
title('Relative injectivity')
xlabel('N_p')
ylabel('rel_p_e_r_m')

legend({'Low Salinity','High Salinity'});
```

8.2 Appendix B: MATLAB-script “LaminarNPart.m”

```
%% Run 1 - low salt concentration
clf
clear
clc

% Conversion factor
Conv_mD = 1.01325.*10e12;                                   % Multiply SI unit with this to get mD (mD/m^2)

% Particles info
dens_p = 1.26;                                              % Fumed alumina density (g/cc)
part_rad_avg = 10e-6;                                       % Expected size of particles in fluid

% Estimate N, the average number of tubings in the core
poro_init = 0.184;                                          % Initial porosity (1/1)
R = 3.81/200;                                               % Core radius (m)
tube_rad_avg = 6e-6;                                        % Average tube radius
N =int32(3/4.*poro_init.*(R/tube_rad_avg).^2);

% Estimating the ammount of tubes on area's (A and B) of the core
a = 80;                                                     % Hit rating a (%)
b = 20;                                                     % Hit rating b (%)

R_a_percent = 80;                                           % Relative radius (%) of entire core cross-section
R_b_percent = 20;                                           % Relative radius (%) of entire core cross-section

R_a = R*(R_a_percent/100);                                  % Middle area with a% hit rating
R_b = R*(R_b_percent/100);                                  % Outer area with b% hit rating

A_a = (R_a^2)*pi;                                           % Area A
A_b = (R^2 - R_a^2)*pi;                                     % Area B

A_a_percent = (A_a)/(A_a+A_b);                              % Area A % of whole core
A_b_percent = (A_b)/(A_a+A_b);                              % Area B % of whole core

N_A = A_a_percent*N;                                        % Ammount of tubes in area A
N_B = A_b_percent*N;                                        % Ammount of tubes in area B

% Lognormal distribution of tubes in the core
m = 1;                                                      % mean value
v = 0.5;                                                    % variance
mu = log((m^2)/sqrt(v+m^2));                                % myu for lognormal function
sigma = sqrt(log(v/(m^2)+1));                               % sigma for lognormal function

rad_tube_A = lognrnd(mu,sigma,[1,N_A]);                     % random number generated from lognormal distribution
rad_tube_B = lognrnd(mu,sigma,[1,N_B]);

rad_tube_A = sort(rad_tube_A);                              % sort the random number from small to large
rad_tube_B = sort(rad_tube_B);

P_A = lognpdf(rad_tube_A,mu,sigma);                         % create distribution function from random number generated
P_B = lognpdf(rad_tube_B,mu,sigma);

rad_tube1_A = rad_tube_A*tube_rad_avg;                      % convert the random number to micrometer size
rad_tube1_B = rad_tube_B*tube_rad_avg;

% Lognormal distribution for the particles has to be moved to the loop...

% Reynolds number
Q = 0.001;                                                  % (m^3/s)
D_core = R*2;                                               % (m)
dyn_visc = 50/1000;                                         % Dynamic viscosity (Pa*s)
kin_visc = dyn_visc./(dens_p*1000);                         % kinematic viscosity (m^2/s)
kin_visc_cSt = kin_visc*10e6;                               % kinematic viscosity centi-stokes (cSt)
A_core = pi()*R.^2;                                         % Area of core (m)
Re = (Q*D_core) / (kin_visc*A_core);                        % Reynolds number


% Absolute permeability without salt precipitation:
k_absi_nosalt_A = (1./(8*((R_a_percent/100)*R).^2))*sum(rad_tube1_A.^4);
k_absi_nosalt_mD_A = k_absi_nosalt_A * Conv_mD;

k_absi_nosalt_B = (1./(8*(R.^2 - ((R_a_percent/100)*R).^2)))*sum(rad_tube1_B.^4);
k_absi_nosalt_mD_B = k_absi_nosalt_B * Conv_mD;

% Estimating delta r, the salt thickness
dry_coef = 1;
D_aq = 1.0974;
X_s = 0.07168;
D_aq1 = 1.14875;
X_s1 = 0.1369;
D_s = 2.16;

salt_sat = (0.85 + dry_coef/3.5)*(D_aq*X_s/D_s);            % solid salt saturation

del_r_A = 2/3*(rad_tube1_A*salt_sat)/dry_coef;              % delta r
del_r_B = 2/3*(rad_tube1_B*salt_sat)/dry_coef;

r_f_A = rad_tube1_A - del_r_A;                              % r - delta_r
r_f_B = rad_tube1_B - del_r_B;

% Calculation of absolute permeability k_abs initial
k_absi_A = (1./(8*((R_a_percent/100)*R).^2))*sum(r_f_A.^4);
k_absi_B = (1./(8*(R.^2 - ((R_a_percent/100)*R).^2)))*sum(r_f_B.^4);

k_absi_mD_A = k_absi_A * Conv_mD;
k_absi_mD_B = k_absi_B * Conv_mD;

k_abs_A = k_absi_A;
```

```
k_abs_B = k_absi_B;

% Defining vectors to be plotted:
k_abs_vector_A = [];
k_abs_vector_A(1,1) = k_absi_mD_A;
k_abs_vector_B = [];
k_abs_vector_B(1,1) = k_absi_mD_B;

rel_perm_vector_A = [];
rel_perm_vector_A(1,1) = 1 - (k_absi_mD_A / k_absi_nosalt_mD_A);
rel_perm_vector_B = [];
rel_perm_vector_B(1,1) = 1 - (k_absi_mD_B / k_absi_nosalt_mD_B);

N_p_vector_A = [];
N_p_vector_A(1,1) = 0;
N_p_vector_B = [];
N_p_vector_B(1,1) = 0;


N_p = 0;                                                % Number of particles in a cubic meter
N_p_step = 100000;                                      % The ammount of particles in each step
j = 1;                                                  % Step counter for plotting the vectors
r_count_A = 0;                                          % Reset counter for when we reduce the tubes N
r_count_B = 0;

while N_p < 1000001
r_f_redu_A = r_f_A;                                     % r_f reduced (used so we can keep our initial r_f values
intact with iterations)
r_f_redu_B = r_f_B;

% Lognormal distribution of particles in the core
%m = 0;                                                 % mean value
%v = 0.25;                                              % variance
mu_part = 0;                                            % myu for lognormal function
sigma_part = 0.5;                                       % sigma for lognormal function

part_dist = lognrnd(mu_part,sigma_part,[1,N_p]);        % random number generated from lognormal distribution
part_dist = sort(part_dist);                            % sort the random number from small to large
P_part = lognpdf(part_dist,mu,sigma);                   % create distribution function from random number generated
part_dist1 = part_dist*part_rad_avg;                    % convert the random number to micrometer size

    i = 1;                                              % Particle "i" that will act upon the system of tubes
    while i < N_p + 1

        randi(100);

        if randi(100)<a || randi(100)==a                % Area A

            elim = randi(N_A);
            if N_p > 0
            elimp = randi(N_p);
            else
            elimp = 1;
            end

            if part_dist1(1,elimp) < r_f_redu_A(1,elim)
            else
            part_dist1(1,elimp) = 0;
            part_dist1 = part_dist1(0~=part_dist1);
            N_p = N_p - 1;

            r_f_redu_A(1,elim) = 0;                     % Eliminate tube where the size of the particle is greater
            r_f_redu_A = r_f_redu_A(0~=r_f_redu_A);     % Eliminating zeros
            N_A = N_A - 1;                              % Reduce the N ammount of tubes when plugged
            r_count_A = r_count_A + 1;
            end

        else                                            % Area B

            elim = randi(N_B);
            if N_p > 0
            elimp = randi(N_p);
            else
            elimp = 1;
            end

            if part_dist1(1,elimp) < r_f_redu_B(1,elim)
            else
            part_dist1(1,elimp) = 0;
            part_dist1 = part_dist1(0~=part_dist1);
            N_p = N_p - 1;

            r_f_redu_B(1,elim) = 0;                     % Eliminate tube where the size of the particle is greater
            r_f_redu_B = r_f_redu_B(0~=r_f_redu_B);     % Eliminating zeros
            N_B = N_B - 1;
            r_count_B = r_count_B + 1;
            end


        end

    i = i + 1;
    end

% Reseting N tubes
N_A = N_A + r_count_A;
N_B = N_B + r_count_B;
N_p = N_p + r_count_A + r_count_B;
r_count_A = 0;
r_count_B = 0;

% Getting values for the vectors, area A
```

```
k_abs_A = (1./(8*((R_a_percent/100)*R).^2))*sum(r_f_redu_A.^4);

k_abs_mD_A = k_abs_A * Conv_mD;
rel_perm_A = 1 - (k_abs_mD_A / k_absi_nosalt_mD_A);

k_abs_vector_A(1,j+1) = k_abs_mD_A;
rel_perm_vector_A(1,j+1) = rel_perm_A;
N_p_vector_A(1,j+1) = N_p;


% Getting values for the vectors, area B
k_abs_B = (1./(8*(R.^2 - ((R_a_percent/100)*R).^2)))*sum(r_f_redu_B.^4);

k_abs_mD_B = k_abs_B * Conv_mD;
rel_perm_B = 1 - (k_abs_mD_B / k_absi_nosalt_mD_B);

k_abs_vector_B(1,j+1) = k_abs_mD_B;
rel_perm_vector_B(1,j+1) = rel_perm_B;
N_p_vector_B(1,j+1) = N_p;


N_p = N_p + N_p_step
j = j + 1;
end



% Run 2 - high salt concentration

% Not reapeating constants

% Estimating delta r, the salt thickness
dry_coef = 1;
D_aq = 1.0974;
X_s = 0.07168;
D_aq1 = 1.14875;
X_s1 = 0.1369;
D_s = 2.16;

salt_sat1 = (0.85 + dry_coef/3.5)*(D_aq1*X_s1/D_s);           % solid salt saturation

del_r_A1 = 2/3*(rad_tube1_A*salt_sat1)/dry_coef;              % delta r
del_r_B1 = 2/3*(rad_tube1_B*salt_sat1)/dry_coef;

r_f_A1 = rad_tube1_A - del_r_A1;                              % r - delta_r
r_f_B1 = rad_tube1_B - del_r_B1;

% Calculation of absolute permeability k_abs initial
k_absi_A1 = (1./(8*((R_a_percent/100)*R).^2))*sum(r_f_A1.^4);
k_absi_B1 = (1./(8*(R.^2 - ((R_a_percent/100)*R).^2)))*sum(r_f_B1.^4);

k_absi_mD_A1 = k_absi_A1 * Conv_mD;
k_absi_mD_B1 = k_absi_B1 * Conv_mD;

% Defining vectors to be plotted:
k_abs_vector_A1 = [];
k_abs_vector_A1(1,1) = k_absi_mD_A1;
k_abs_vector_B1 = [];
k_abs_vector_B1(1,1) = k_absi_mD_B1;

rel_perm_vector_A1 = [];
rel_perm_vector_A1(1,1) = 1 - (k_absi_mD_A1 / k_absi_nosalt_mD_A);
rel_perm_vector_B1 = [];
rel_perm_vector_B1(1,1) = 1 - (k_absi_mD_B1 / k_absi_nosalt_mD_B);

N_p_vector_A1 = [];
N_p_vector_A1(1,1) = 0;
N_p_vector_B1 = [];
N_p_vector_B1(1,1) = 0;


N_p1 = 0;                                                     % Number of particles in a cubic meter
N_p_step1 = 100000;                                           % The ammount of particles in each step
j = 1;                                                        % Step counter for plotting the vectors
r_count_A = 0;                                                % Reset counter for when we reduce the tubes N
r_count_B = 0;

while N_p1 < 1000001
r_f_redu_A1 = r_f_A1;                                         % r_f reduced (used so we can keep our initial r_f values intact
with iterations)
r_f_redu_B1 = r_f_B1;

% Lognormal distribution of particles in the core
%m = 0;                                                       % mean value
%v = 0.25;                                                    % variance
mu_part = 0;                                                  % myu for lognormal function
sigma_part = 0.5;                                             % sigma for lognormal function

part_dist = lognrnd(mu_part,sigma_part,[1,N_p1]);             % random number generated from lognormal distribution
part_dist = sort(part_dist);                                  % sort the random number from small to large
P_part = lognpdf(part_dist,mu,sigma);                         % create distribution function from random number generated
part_dist1 = part_dist*part_rad_avg;                          % convert the random number to micrometer size

    i = 1;                                                    % Particle "i" that will act upon the system of tubes
    while i < N_p1 + 1

randi(100);

        if randi(100)<a || randi(100)==a                      % Area A

        elim = randi(N_A);
        if N_p1 > 0
        elimp = randi(N_p1);
```

```
        else
        elimp = 1;
        end
            if part_dist1(1,elimp) < r_f_redu_A1(1,elim)
            else
            part_dist1(1,elimp) = 0;
            part_dist1 = part_dist1(0~=part_dist1);
            N_p1 = N_p1 - 1;

            r_f_redu_A1(1,elim) = 0;                        % Eliminate tube where the size of the particle is greater
            r_f_redu_A1 = r_f_redu_A1(0~=r_f_redu_A1);      % Eliminating zeros
            N_A = N_A - 1;
            r_count_A = r_count_A + 1;
            end

        else                                                % Area B

        elim = randi(N_B);
        if N_p1 > 0
        elimp = randi(N_p1);
        else
        elimp = 1;
        end
            if part_dist1(1,elimp) < r_f_redu_B1(1,elim)
            else
            part_dist1(1,elimp) = 0;
            part_dist1 = part_dist1(0~=part_dist1);
            N_p1 = N_p1 - 1;

            r_f_redu_B1(1,elim) = 0;                        % Eliminate tube where the size of the particle is greater
            r_f_redu_B1 = r_f_redu_B1(0~=r_f_redu_B1);
            N_B = N_B - 1;
            r_count_B = r_count_B + 1;
            end



        end

    i = i + 1;
    end

% Reseting N tubes
N_A = N_A + r_count_A;
N_B = N_B + r_count_B;
N_p1 = N_p1 + r_count_A + r_count_B;
r_count_A = 0;
r_count_B = 0;

% Getting values for the vectors, area A
k_abs_A1 = (1./(8*((R_a_percent/100)*R).^2))*sum(r_f_redu_A1.^4);

k_abs_mD_A1 = k_abs_A1 * Conv_mD;
rel_perm_A1 = 1 - (k_abs_mD_A1 / k_absi_nosalt_mD_A);

k_abs_vector_A1(1,j+1) = k_abs_mD_A1;
rel_perm_vector_A1(1,j+1) = rel_perm_A1;
N_p_vector_A1(1,j+1) = N_p1;


% Getting values for the vectors, area B
k_abs_B1 = (1./(8*(R.^2 - ((R_a_percent/100)*R).^2)))*sum(r_f_redu_B1.^4);

k_abs_mD_B1 = k_abs_B1 * Conv_mD;
rel_perm_B1 = 1 - (k_abs_mD_B1 / k_absi_nosalt_mD_B);

k_abs_vector_B1(1,j+1) = k_abs_mD_B1;
rel_perm_vector_B1(1,j+1) = rel_perm_B1;
N_p_vector_B1(1,j+1) = N_p1;


N_p1 = N_p1 + N_p_step1
j = j + 1;
end

%% Plotting
figure('Name','Size = 10e-6 (m), Porosity = 0.184')
subplot(2,2,1)
plot(N_p_vector_A,k_abs_vector_A,'b',N_p_vector_A1,k_abs_vector_A1,'r')
grid on
title('Absolute permeability, Area A')
xlabel('N_p')
ylabel('k_a_b_s mD')

subplot(2,2,2)
plot(N_p_vector_B,k_abs_vector_B,'b',N_p_vector_B1,k_abs_vector_B1,'r')
grid on
title('Absolute permeability, Area B')
xlabel('N_p')
ylabel('k_a_b_s mD')

subplot(2,2,3)
plot(N_p_vector_A,rel_perm_vector_A,'b',N_p_vector_A1,rel_perm_vector_A1,'r')
grid on
title('Relative injectivity, Area A')
xlabel('N_p')
ylabel('rel_p_e_r_m')

subplot(2,2,4)
plot(N_p_vector_B,rel_perm_vector_B,'b',N_p_vector_B1,rel_perm_vector_B1,'r')
grid on
title('Relative injectivity, Area B')
xlabel('N_p')
```

```
ylabel('rel_p_e_r_m')

legend({'Low Salinity','High Salinity'});
%% Optional plotting
figure('Name','insert name here')
subplot(2,1,1)
plot(rad_tube1_A,P_A)
grid on
title('Lognormal distribution, Area A')
xlabel('r_i');
ylabel('P');

subplot(2,1,2)
plot(rad_tube1_B,P_B)
grid on
title('Lognormal distribution, Area B')
xlabel('r_i');
ylabel('P');
```