A new two-phase rock typing approach: Using wetting-phase relative permeability and critical pore size

by

Brandon A. Yokeley

B.Sc., Appalachian State University, 2019

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Geology
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Approved by:

Major Professor
Dr. Behzad Ghanbarian

Copyright

Abstract

Rock typing, the methodology of grouping rocks based on their mineralogical, hydraulic, and/or petrophysical similarities has many applications in reservoir engineering, characterization, and simulation. Grouping rocks based on single-phase data has been widely discussed in the literature, while two-phase rock typing methodologies are limited. Single-phase rock typing methods generally identify rock types using porosity and permeability measurements based on similar characteristic pore sizes. To address the effect of wettability, ignored in single-phase rock typing, the present study focuses on classifying rock types using two-phase flow data. Using concepts from critical-path analysis (CPA) we propose a new rock typing methodology based on wetting-phase relative permeability curve, k_{rw}, critical pore radius, r_c, and effective wetting-phase saturation, S_e. For this purpose, we convert the wetting-phase relative permeability curves, $S_w - k_{rw}$, to $S_e - r_c$ curves. We utilize a curve clustering method to identify representative rock types. To assess the proposed rock typing approach, we first created a large petrophysical dataset using pore network simulations, that covered a comprehensive range of pore size distributions, contact angles, pore coordination numbers, pore shape distributions, and clay contents. Overall, 240 pore networks were generated. We also simulated two-phase flow in six additional pore networks based on properties of Berea, Mt. Simon, and Fontainebleau sandstones. Results showed that there exist twelve unique rock types in our dataset. However, using single-phase rock typing techniques, we found a different number of rock types ranging between eight and fifteen depending on the methods applied. The discrepancies in the results of single- and two-phase rock typing approaches highlighted the importance of classifying rocks using two-phase flow data.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to acknowledge the never ending support and direction given by my advisor, Dr. Ghanbarian. His insight and criticism has been much appreciated. I would also like to thank my significant other as she has had to listen to my never ending babel about rock typing and code. Her mental and emotional help has been priceless. Finally, I would like to thank the Department of Geology at Kansas State University for funding this project and giving me the opportunity to work on this project and acquire my masters degree.

Chapter 1

Introduction

The process of classifying reservoir rocks into similar groups is known as rock typing in the petroleum industry. Rock typing based on their fluid flow characteristics is vital in accurately modeling oil and gas reservoirs. Being able to precisely group rocks based on their fluid flow behavior is essential in setting up reservoir models. One of the pioneering rock typing works was conducted by Archie (1950) who states

> "Though permeable rocks are, by nature, heterogeneous, their characteristics follow definite trends when considering a formation as a whole . . . a formation whose parts have been deposited under similar conditions and have undergone similar processes of later weathering, cementation or re-solution [will be considered a rock type]" (Archie, 1950)

With advances in technology and our knowledge of flow in porous media, rock typing has been substantially progressed and applied in the oil and gas industry. Rock typing can be performed using different ways, due to the complex nature of oil and gas reservoirs. The most common avenues of approach for rock typing found within the literature, in no particular order, are as follows:

- Using the local stratigraphy to classify rocks into separate rock types, such as grouping similar stratigraphic units together into one rock type (Gregorio et al., 2020; Gunter et al., 1997; Lucia, 1995; Rushing et al., 2008).

- Classifying rocks based on their petrological and diagenetic similarities, particularly when working with carbonate reservoirs (Aliakbardoust and Rahimpour-Bonab, 2013; Farshi et al., 2019; Gunter et al., 1997; Rushing et al., 2008).

- Grouping rocks with similar fluid flow or petrophysical properties into representative rock types, using data from cores and/or well logs (Aliakbardoust and Rahimpour-Bonab, 2013; Compan et al., 2016; Farshi et al., 2019; Ghanbarian et al., 2019; Hamon and Bennes, 2004; Mirzaei-Paiaman and Ghanbarian, 2020; Mirzaei-Paiaman et al., 2018, 2019b; Rushing et al., 2008).

Rock typing can be carried out with any number of these approaches, depending on data availability, by combining methods typically used (Aliakbardoust and Rahimpour-Bonab, 2013; Gregorio et al., 2020; Gunter et al., 1997; Hamon and Bennes, 2004; Hollis et al., 2010; Rushing et al., 2008; Skalinski and Kenter, 2015). When core data are available, petrological and petrophysical data can be used to group rocks into their representative rock types (Gunter et al., 1997; Rushing et al., 2008). While when well logs are present, stratigraphic and petrophysical methods can be used to type rocks into their representative rock types (Archie, 1950; Rushing et al., 2008). When typing rocks using petrophysical data, there is two different types of data available, single-phase and two-phase flow measurements. Single-phase data indicate that there is only one fluid within a sample, water, gas, or oil for instance. While two-phase data mean that there are two fluids present e.g., oil-water, oil-gas, or water-gas. Within the literature, the single-phase rock typing is much more common than the two-phase rock typing, probably because porosity and permeability are routinely measured in oil/gas explorations. However, single-phase flow measurements do not capture any information about the wettability and contact angle, particularly in mix-wet reservoirs. Thus, this study will focus on using two-phase petrophysical data to group rocks into their representative types.

In this study, we present a new approach for two-phase rock typing based on concepts of critical-path analysis and by determining critical pore sizes at different effective saturations to classify rocks. Due to limitations acquiring experimental data, in any form, in our study

we first generate a large petrophysical database that covers a comprehensive types of rocks. We validate our proposed rock typing method using a curve clustering method to classify rocks that share similar critical pore sizes within the generated database.

Chapter 2

Literature Review

The literature on rock typing is extensive and covers a wide variety of methods and reservoirs, from tight-gas sandstones to highly heterogeneous carbonate reservoirs, see e.g., (Aliakbardoust and Rahimpour-Bonab, 2013; Amaefule et al., 1993; Archie, 1950; Compan et al., 2016; Farshi et al., 2019; Gregorio et al., 2020; Gunter et al., 1997; Hamon and Bennes, 2004; Hollis et al., 2010; Kolodzie, 1980; Lucia, 1995; Mirzaei-Paiaman and Ghanbarian, 2020; Mirzaei-Paiaman et al., 2018, 2019b; Skalinski and Kenter, 2015; Winsauer et al., 1952) and references therein. The basis of different rock typing methods is to detect representative rock types within a reservoir. In this chapter, several common rock typing methods are discussed with a critical focus on using petrophysical data to determine rock types using either single-phase or two-phase flow measurements. Further comprehensive reviews on rock typing can be found in the following references: Faramarzi-Palangar and Mirzaei-Paiaman (2020a); Ghanbarian et al. (2019); Kadkhodaie and Kadkhodaie (2018); Michel and Bruno (2014); Rushing et al. (2008).

2.1 Single-Phase Rock Typing

Archie's Law

Rock typing using single-phase data is much more common in the literature compared to two-phase rock typing. As stated in Chapter 1, one of the first forms of rock typing based on electrical resistivity and porosity data was developed by Archie (1950) and later modified by Winsauer et al. (1952) to create the following well known equation called Archie's law:

$$F = a\phi^{-m} \tag{2.1}$$

This method utilizes the relationship between formation factor (F) and porosity (ϕ) to classify rocks into separate rock types. Building upon this, Archie's law also quantifies how rocks are well cemented using Archie's cementation exponent (m). Rocks that are more well consolidated and cemented exhibit higher m values. Thus, carbonates will generally show greater cementation exponents compared to unconsolidated sandstones (Müller-Huber et al., 2015; Porter and Carothers, 1971). However, Archie's law has limitations, particularly in highly heterogeneous or vuggy carbonates (Müller-Huber et al., 2015). Such limitation with Archie's law led to the development of a variety of other rock typing methods; most common methods are presented in the following.

Winland & Pittman Equations

A large number of rock typing studies is based on the Winland equation (Ghanbarian et al., 2019; Kolodzie, 1980; Rushing et al., 2008). This method was originally developed by H.D. Winland to aid in the calculation of pay zone in hydrocarbon reservoirs. He proposed that there is a relationship between the pore throat size, and the porosity - permeability relationship (Kolodzie, 1980). Using mercury-intrusion capillary pressure (MICP) curves, one can calculate the thirty-fifth percentile of the MICP curve (r_{35}), and then relate that to the permeability and porosity of a rock sample (Eq. 2.2).

$$log(r_{35}) = 0.732 + 0.588 * log(k) - 0.864 * log(\phi) \tag{2.2}$$

Using well-log data from the Spindle Field in Colorado, USA, Kolodzie (1980) modified the original Winland equation (Eq. 2.2) and established the following relationship

$$log(r_{35}) = 0.9058 + 0.5547 * log(k) - 0.90338 * log(\phi) \tag{2.3}$$

Both the original Winland equation (Eq. 2.2) and the modified Winland equation (Eq. 2.3) attempt to correlate pore throat sizes to permeability and porosity, albeit with different numerical prefactors. However, the differences between these two equations, using petrophysical data from two different well fields, illustrate the drawbacks of these empirical formulas, as the constants in these equations must be modified depending on the data set used. Furthermore, there is not a general agreement within the literature that r_{35} is the optimal pore size for calculating permeability (Ghanbarian et al., 2019; Pittman, 1992; Riazi, 2018).

Similar to Winland, Pittman (1992) proposed a set of empirical relationships to relate pore throat size and permeability to MCIP data. However, Pittman (1992) used pore throat radius at different mercury saturations (Table 2.1). This set of equations are based on a range of pore throat radii values, from r_{10} to r_{75}.

Table 2.1: *Pittman equations used to relate pore throat radii determined from MICP curves to porosity and permeability.*

Equation	R^2
$\log(r_{10}) = 0.459 + 0.500 \log(\mathrm{k}) - 0.385 \log(100\phi)$	0.901
$\log(r_{15}) = 0.333 + 0.509 \log(\mathrm{k}) - 0.344 \log(100\phi)$	0.919
$\log(r_{20}) = 0.218 + 0.519 \log(\mathrm{k}) - 0.303 \log(100\phi)$	0.926
$\log(r_{25}) = 0.204 + 0.531 \log(\mathrm{k}) - 0.350 \log(100\phi)$	0.926
$\log(r_{30}) = 0.215 + 0.547 \log(\mathrm{k}) - 0.420 \log(100\phi)$	0.923
$\log(r_{35}) = 0.255 + 0.565 \log(\mathrm{k}) - 0.523 \log(100\phi)$	0.918
$\log(r_{40}) = 0.360 + 0.582 \log(\mathrm{k}) - 0.680 \log(100\phi)$	0.918
$\log(r_{45}) = 0.609 + 0.608 \log(\mathrm{k}) - 0.974 \log(100\phi)$	0.913
$\log(r_{50}) = 0.778 + 0.626 \log(\mathrm{k}) - 1.205 \log(100\phi)$	0.908
$\log(r_{55}) = 0.948 + 0.632 \log(\mathrm{k}) - 1.426 \log(100\phi)$	0.900
$\log(r_{60}) = 1.096 + 0.648 \log(\mathrm{k}) - 1.666 \log(100\phi)$	0.893
$\log(r_{65}) = 1.372 + 0.643 \log(\mathrm{k}) - 1.979 \log(100\phi)$	0.876
$\log(r_{70}) = 1.664 + 0.627 \log(\mathrm{k}) - 2.314 \log(100\phi)$	0.862
$\log(r_{75}) = 1.880 + 0.609 \log(\mathrm{k}) - 2.626 \log(100\phi)$	0.820

There are a number of disadvantages with such empirical models. The first of which is it is not clear whether r_{35} refers to radius or diameter. Kolodzie (1980) uses r_{35} to refer to pore throat size of the 35th percentile on the MCIP curve, while Pittman (1992) use r_{35} as the pore throat radii. These inconsistencies in terminology drastically affect the permeability estimation when using this set of equations. The second disadvantage is their applications to unconventional reservoirs such as tight-gas sandstones and shales are questionable. Both Winland and Pittman equations overestimate permeability in tight reservoir rocks (Ghanbarian et al., 2019; Rezaee et al., 2012; Rushing et al., 2008). This illustrates the issues with using empirically derived rock typing methods, and highlights the extreme dependence on

data used to develop them.

Reservior Quality Index & Hydraulic Flow Units

Another popular method alongside the Winland Equation to type rocks is using flow zone indicators (FZI) and the reservoir quality index (RQI) to categorize rocks into their representative hydrualic flow units, or HFU's (Amaefule et al., 1993; Farshi et al., 2019; Kadkhodaie and Kadkhodaie, 2018; Mirzaei-Paiaman et al., 2015; Riazi, 2018). Amaefule et al. (1993) focused on determining reservoir quality using the peterophysical properties of a reservoir, taking particular notice to pore geometry and the permeability - porosity ratio. Their method based on the generalized Konzeny-Carmen equation resulted in the following equation:

$$RQI = 0.0314\sqrt{\frac{k}{\phi_e}} \tag{2.4}$$

where k is permeability in milldarcy, and ϕ_e is fractional effective porosity. One may calculate the void ratio (ϕ_z), the ratio of pore volume to grain volume, to determine the flow zone indicator (FZI)

$$FZI = \frac{1}{\sqrt{F_s}\tau S_{gv}} = \frac{RQI}{\phi_z} \tag{2.5}$$

This equation includes the shape factor (F_s), toroturosity (τ), and surface area per unit grain volume (S_{gv}). From the prior two equations, Amaefule et al. (1993) then developed the following RQI/FZI relationship that is ultimately used to classify hydraulic flow units

$$log(RQI) = log(\frac{\phi}{1-\phi}) + log(FZI) \tag{2.6}$$

This set of equations is then used to classify rocks that are plotted on a log-log plot of ϕ_z-RQI (Fig. 2.1). Rocks that have similar FZI values fall on a straight line with unit slope (Amaefule et al., 1993), while those with different FZI values fall on different parallel lines that also express a unit slop. These parallel lines indicate rocks that share similar pore throat characteristics, thus forming a single HFU.

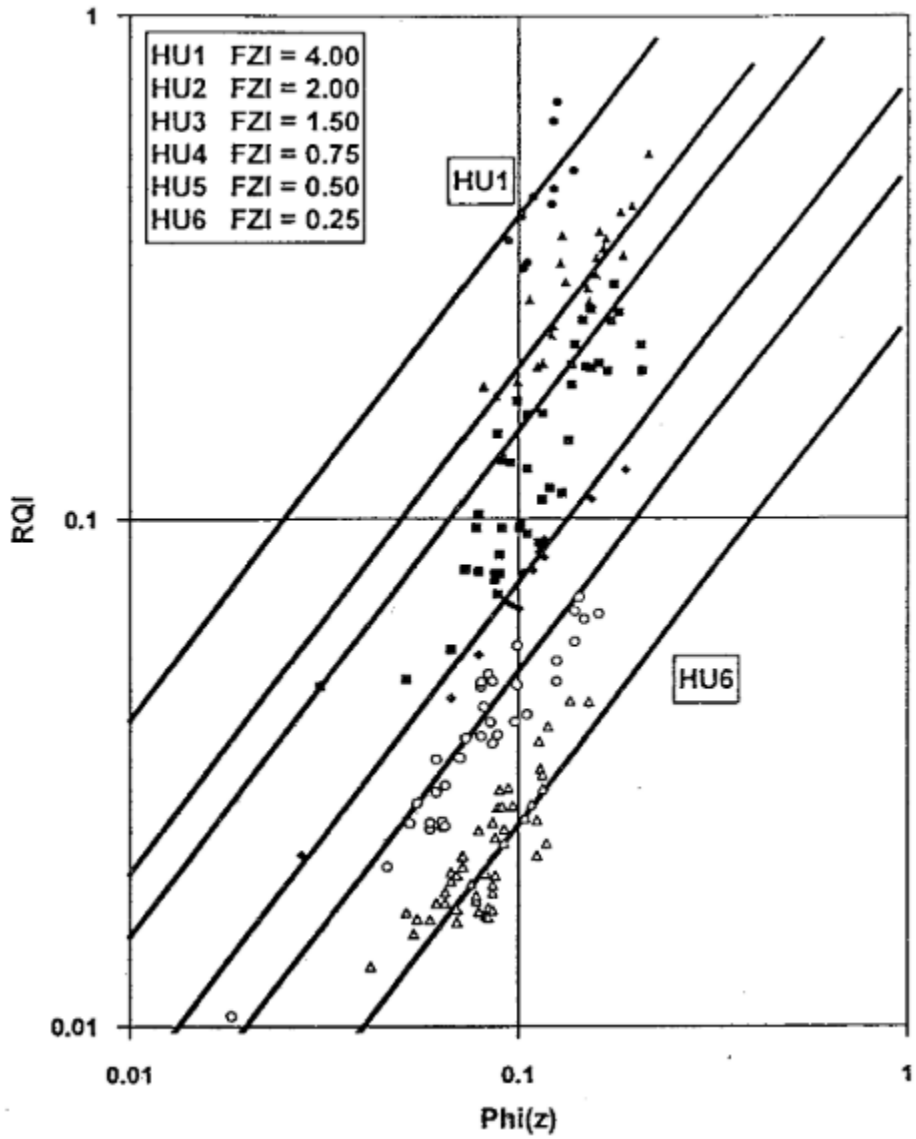

Figure 2.1: *Illustration of how hydraulic flow units are determined using RQI and FZI (after Amaefule et al. (1993)). Rocks that share similar flow characteristics will fall along the same unit slope line and be classifed as a single hydraulic flow unit.*

While this method was revolutionary at the time, more recent studies have pointed to the flaws of using this method. For example, even the generalized Konzeny-Carmen equation does not stand valid in highly complex or heterogeneous rocks (Ghanbarian et al., 2019; Liu et al., 2019; Mirzaei-Paiaman et al., 2015). Based on the fact that the generalized Konzeny-Carmen equation can not be accurately applied to media with broad pore size distributions,

the FZI method is only reliable in clean sandstones.

Recent studies by Mirzaei-Paiaman et al. (2019a, 2015) showed that a modified FZI method can be used for more accurate rock typing. This modified FZI method, referred to as FZI* or FZI-Star by Mirzaei-Paiaman et al. (2019a, 2015), aims to more accurately group rocks that share similar pore geometry by classifying HFU's as rocks with distinct effective or mean hydraulic radii versus classifying HFU's using the permeability - porosity ratio.

Using the original form of the Konzeny-Carmen equation instead of the its generalized version, along with the equation originally developed by Amaefule et al. (1993) (Eq. 2.6), Mirzaei-Paiaman et al. (2015) found the following relationship between mean hydraulic radius (r_{mh}), turoturosity (τ) and shape factor of the pores (F_s)

$$FZI^* = \frac{r_{mh}}{\tau\sqrt{F_s}} \tag{2.7}$$

Using this equation Mirzaei-Paiaman et al. (2015) showed that on the log-log plot of $0.0314\sqrt{k}$-$\sqrt{\phi}$, rocks that are within the same HFU follow the same straight line with unit slope. Due to the nature of this equation, Mirzaei-Paiaman et al. (2015) argued that this is a more accurate way to calculate representative HFU's as this method accounts for all the pore geometry-related parameters in the Konzeny-Carmen equation, mean hydraulic radius, turoturosity, and shape factor. However, their approach still does not incorporate the effect of wettability on rock typing. In the evaluation of the FZI* method on a highly heterogeneous reservoir, Mirzaei-Paiaman et al. (2015) found that their approach correctly identified HFU, while using FZI failed to recognize individual HFU's. Further verification of this model, FZI*, was conducted by Mirzaei-Paiaman et al. (2019a) in which they stated that FZI* better recognized rock types when compared with other methods such as the Winland r_{35} and FZI methods.

Characteristic Pore Size

Due to empirical nature and/or limitations of the previous rock typing methods, Ghanbarian et al. (2019) developed a new methodology for rock typing based on grouping rocks with

characteristic pore sizes. Ghanbarian et al. (2019) applied the following equation to group rocks based on their characteristic pore sizes:

$$k = \frac{l_c^2}{cF} \tag{2.8}$$

where l_c is the characteristic length scale, F is the formation factor, and c is a constant whose value ranges between 8 and 226, depending on the theory applied (Ghanbarian et al., 2019). The inclusion of both permeability and formation factor in this relationship is pivotal to the basis of this rock typing procedure, as both parameters depend on the dynamic topology of rock samples. In addition to that, the inclusion of formation factor inherently includes the effect of the cementation exponent (m) in Archie's law (Eq. 2.1).

Following Johnson et al. (1986) and their theory, Ghanbarian et al. (2019) replaced the characteristic length scale l_c with the characteristic pore size Λ (Johnson et al., 1986), and set $c = 8$. They then plotted permeability versus the reciprocal of formation factor and next detected rocks with similar Λ values. Ghanbarian et al. (2019) demonstrated that the cementation exponent m plays a nontrivial role in the selection of rock types as shown in Fig. 2.2. Thus, to incorporate the effect of the cementation exponent into their methodology, they plot samples based on $1/F$-k relationship, and group rocks with similar Λ values into representative rock types. By comparing their method of grouping to that of the Winland equation, they found that Λ more accurately characterizes rock types.

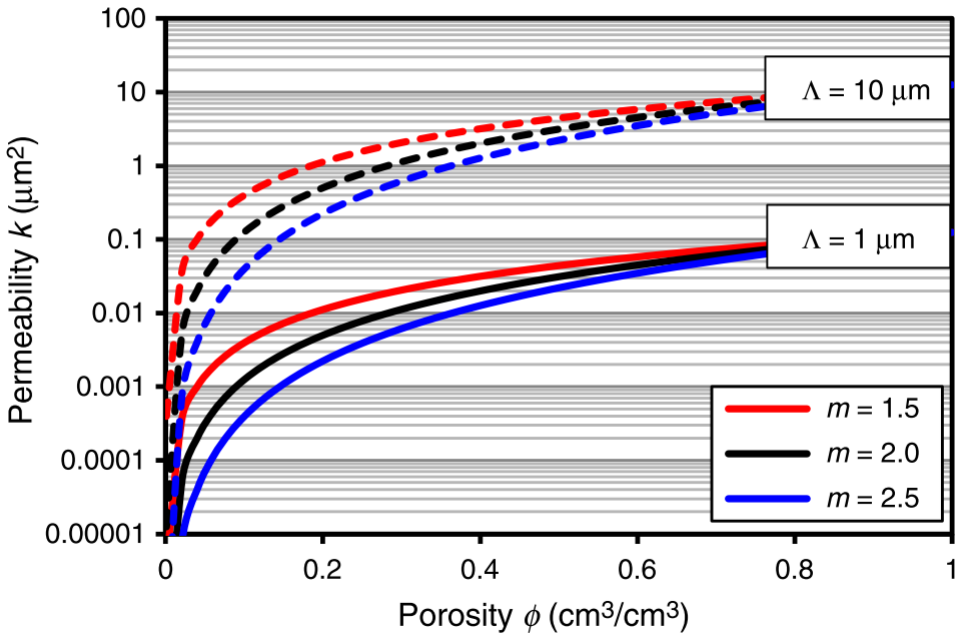

Figure 2.2: *The impact of the cementation exponent on different* Λ *values for* k *plotted versus phi (after Ghanbarian et al. (2019)). Rocks with the same* Λ *values but different* m *values will display different clusters when comparing porosity against permeability. Thus, the use of analyzing the data set using the* $1/F$*-*k *relationship.*

Even though the Ghanbarian et al. (2019) method has a solid theoretical basis compared to an empirical method to classify rock types, it still has limitations in terms of applications, particularly to mix-wet formations. Given that neither permeability nor formation factor incorporates the impact of wettability, care must be taken. If one wants to attempt this method on mixed-wet samples, the inflection point (l_{inf}) of the capillary pressure curve can be converted into Λ. Furthermore, the Ghanbarian et al. (2019) method does not differentiate among rocks with multi-modal pore size distributions. It groups uni- and multi-modal rocks with similar Λ values in the same type.

2.2 Two-Phase Rock Typing

Two-phase rock typing is a new method in industry. Even in the literature, applications from two-phase rock typing are limited Compan et al. (2016); Hamon and Bennes (2004); Mirzaei-Paiaman and Ghanbarian (2020); Mirzaei-Paiaman et al. (2019b). While single-phase rock typing focuses on the static behaviors of reservoirs, two-phase rock typing is based on the dynamic behavior of formations. Due to the complex nature of fluid flow in reservoirs, particularly when attributing wettability and pore geometry of a rock, two-phase rock typing can be considered as a more accurate way to classify rocks. However, access to two-phase data in hydrocarbon reservoirs can be costly and/or difficult. That might be the reason of limited applications of two-phase rock typing studies in the literature.

Typical single-phase rock typing methods ignore the effect of wettability on flow and transport. Ignoring that can lead to inaccurate representation of rock types because rocks with dissimilar contact angles may be classified in the same group. It is well documented in the literature that the effect of wettability on relative permeability is non-trivial (Anderson, 1987; Blunt, 1997; Dicarlo et al., 2000; Li et al., 2005; Mahmud et al., 2007; Xu et al., 2014). Accordingly, single-phase rock typing can lead to inaccurate grouping of rocks. However, two-phase rock typing inherently includes the influence of wettability through two-phase flow measurements, such as water relative permeability or capillary pressure curves (An et al., 2016; Mirzaei-Paiaman et al., 2019b; Nishiyama and Yokoyama, 2017).

True Effective Mobility Function

A new model for two-phase rock typing was recently proposed by Mirzaei-Paiaman et al. (2019b), Faramarzi-Palangar and Mirzaei-Paiaman (2020b) and Mirzaei-Paiaman and Ghanbarian (2021), in which a new parameter named True Effective Mobility Function, or TEM, is used to classify rocks into petrophysical dynamic rock types, or PDRT's. Using relative permeability data along with additional petrophysical properties those authors aimed to minimize uncertainties in reservoir simulation models. Based on Darcy's law (Eq. 2.9), Mirzaei-Paiaman et al. (2019b) proposed the TEM function (Eq. 2.10) to classify rocks with

similar TEM values:

$$u_\alpha = -\frac{k * k_{r\alpha}}{\mu_\alpha} * \Delta P_\alpha \tag{2.9}$$

$$TEM_\alpha = \frac{k * k_{r\alpha}}{\phi * \mu_\alpha} \tag{2.10}$$

where k is permeability, ϕ is porosity, and $k_{r\alpha}$ and μ_α are relative permeability and dynamic viscosity of the the selected fluid phase, respectively.

Within the TEM function framework, samples with similar dynamic fluid flow characteristics would have similar TEM curves. The TEM curves represent the capability of a sample to transmit fluids. This means that samples with greater TEM values on the S_w-TEM plot would have higher quality, as they exhibit better fluid flow characteristics. Studies by Mirzaei-Paiaman and his coworkers show that when strictly using water relative permeability or capillary pressure data, rocks with similar fluid flow characteristics can not be accurately grouped. This is shown by data presented in Fig. 2.3. Experimental data from the Bangestan Group exhibit significant overlap when strictly looking at the water relative permeability data. However, when using the TEM function given by equation 2.10, samples were partitioned into representative PDRT's.

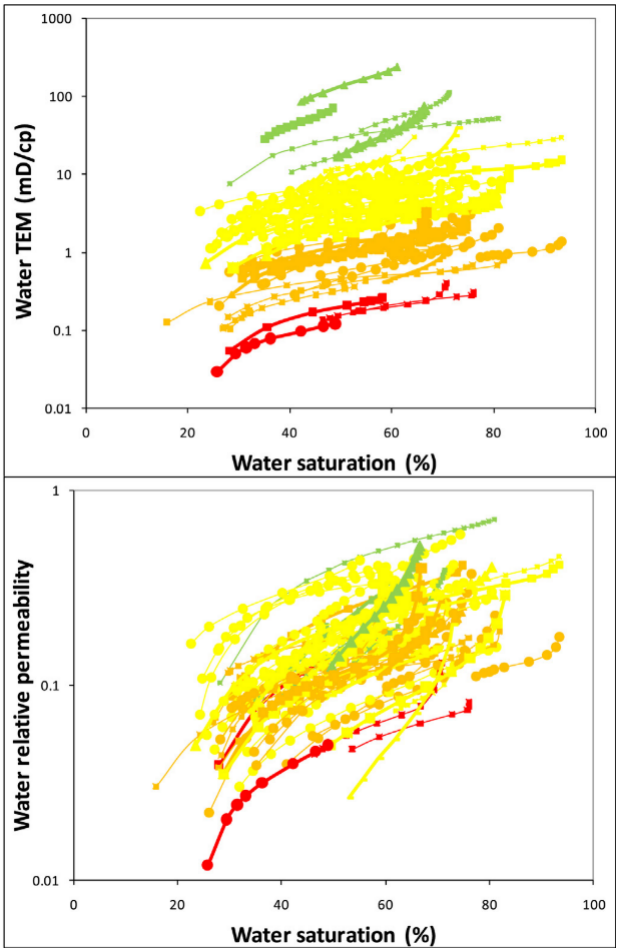

Figure 2.3: *TEM and water relative permeability curves from the Ilam and Sarvak carbonate Formations within the Bangestan Group. Top most graph shows four distinct PDRT's found using equation 2.10 while the bottom most graph shows the significant overlap of these distinct PDRT's. Thus, indicating the inaccuracy of the using only water relative permeability curves to depict PDRT's. Figure modified from Mirzaei-Paiaman et al. (2019b)*

More recently, Mirzaei-Paiaman and Ghanbarian (2021) showed that plotting TEM curves against imbibed water saturation instead of water saturation resulted in more accurate rock typing. This idea of using imbibed water saturation was also extended for rock typing using capillary pressure curves, as discussed in the following.

Averaging Capillary Pressure Curves

Two-phase rock typing using capillary pressure curves was proposed by Mirzaei-Paiaman and Ghanbarian (2020). In their study, those authors argued that performing rock typing by comparing only primary drainage capillary pressure curves is "physically meaningless and not supported." Furthermore, using only relative permeability curves can also lead to inaccurate representation of rock types, although relative permeability curves do completely represent the dynamic fluid properties of a rock. They proposed a different method of using capillary pressure curves under imbibition and/or secondary drainage, along with imbibed water saturation (Eq. 2.11) to carry out a more accurate rock typing:

$$S_{W,I} = S_W - S_{WC} \tag{2.11}$$

Imbibition and/or secondary drainage data provide a better estimate of the fluid flow characteristics of a rock, given that rock with similar primary drainage curves could have different fluid flow characteristics depending on pore topology and wettability. By plotting both secondary drainage and imbibition capillary pressure curves against imbibed water saturation, all capillary pressure curves would start at the same point. This means that rocks with similar oil recoveries under imbibition, and similar water recoveries under secondary drainage, would collapse into one another. higher quality reservoir rocks would be shifted farther right on the $S_{w,I}$-Capillary pressure plot (Fig. 2.4).

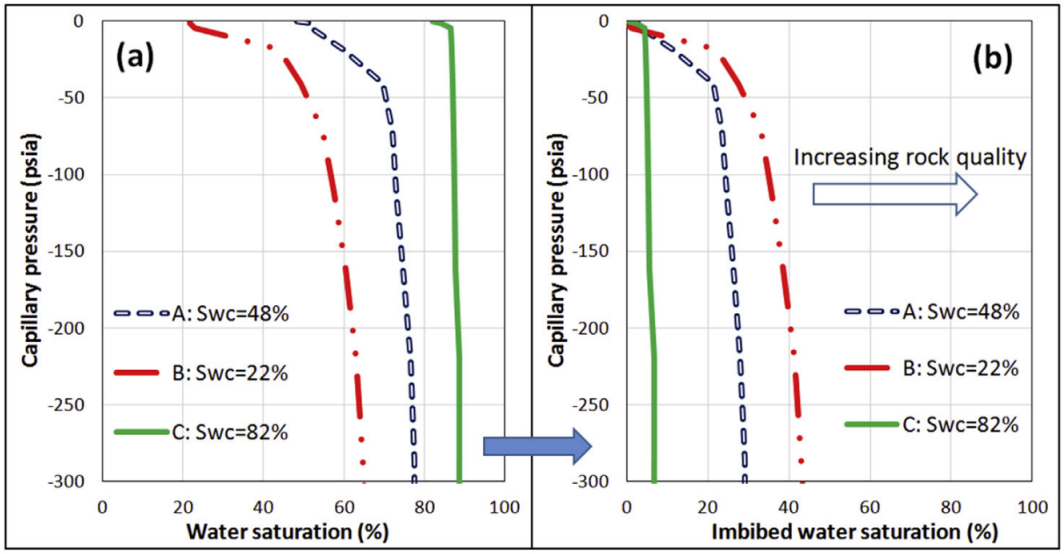

Figure 2.4: *Demonstration of how the rock typing procedure proposed in Mirzaei-Paiaman and Ghanbarian (2020) functions. With curves all starting at a central point and higher quality reservoir rocks being further right on the imbibed water saturation - capillary pressure curves (after Mirzaei-Paiaman and Ghanbarian (2020))*

Continuing their study of capillary pressure curves, Mirzaei-Paiaman and Ghanbarian (2020) proposed a new methodology for representing rock types in commercially available reservoir simulations, as there are instances when rock typing completed using primary drainage capillary pressure curves are not consistent with rock types determined from imbibition and/or secondary drainage capillary pressure curves (Mirzaei-Paiaman and Ghanbarian, 2020). This proposed methodology aims to alleviate this inconsistency by incorporating the utilization of imbibition and/or secondary drainage capillary pressure curves. By checking the predetermined rock types from primary drainage capillary pressure curves with the imbibition and/or secondary drainage capillary pressure curves, rock types can be better defined, such that rocks with similar pore topology and wettability, i.e. similar imbibition and/or secondary drainage capillary pressure curves, will be classified into the same type.

2.3 General Overview of Rock Typing

As stated earlier, there are numerous ways to conduct rock typing based upon data availability. Single-phase rock typing is much more common compared to two-phase one, although it has limitations, particularly when empirical-based models such as the Winland equation and Pittman equations are applied (Ghanbarian et al., 2019). Meanwhile, other methods such as RQI/FZI may not accurately detect rock types or hydraulic flow units, as they do not completely encompass the effects of pore geometry (Mirzaei-Paiaman et al., 2019a, 2015). Thus, new approaches based on classifying hydraulic flow units were developed in the form of FZI*, which better encompass pore geometry (Mirzaei-Paiaman et al., 2019a, 2015). However, single-phase rock typing methods still do not include the effect of wettability. Given that contact angle in reservoirs might spatially vary, single-phase rock typing approaches can not fully capture the complex nature of hydrocarbon reservoirs (Hamon and Bennes, 2004).

The literature lacks a solid theoretical method of two-phase rock typing addressing both the effect of pore geometry and wettablility, although it requires more petrophysical measurements than single-phase rock typing methods. Nonetheless, two-phase rock typing approaches provide a more accurate way to detect type rocks, as they utilize the dynamic fluid flow characteristics of a rock (Mirzaei-Paiaman et al., 2019b).

In this study, we propose a new and theoretical way to classify rocks based on their two-phase petrophysical properties. Applying concepts of critical path analysis from statistical physics and generalizing the theory developed by Ghanbarian et al. (2019), we construct a novel methodology to classify rock types using critical pore sizes, water relative permeability curves, critical water saturation, and effective saturation. This proposed rock typing method is explained, in detail, in the proceeding chapters.

Chapter 3

Methods and Materials

To develop the proposed rock typing method based on two-phase flow characteristics, we used pore-network modeling to simulate single- and two-phase flow in porous media with a wide range of pore-scale heterogeneity. In what follows, we first explain the pore-scale simulations, and then the rock typing method developed in this study.

3.1 Pore-network simulations

3.1.1 Synthetic porous media

To carry out pore-scale simulations, we used the open-source pore-network model developed by Valvatne (2004), which generates pore networks of size $n \times n \times n$ in which n represents the number of pore bodies in the three directions. For the sake of simplicity, we use n to indicate the size of the pore networks, which were constructed by randomly distributing the pore-throats radii r that follow the truncated Weibull distribution

$$r = (r_{\max} - r_{\min}) \left[-\delta \ln x[1 - \exp(-1/\delta)] + \exp(-1/\delta)\right]^{1/\gamma} + r_{\min} \tag{3.1}$$

where δ and γ are the Weibull distribution parameters, x is a random number distributed uniformly between zero and one, and $r_{\min}$ and $r_{\max}$ are the smallest and largest pore-throat

radii in the network. The pore-body radius r_b is related to the pore-throat radius r through the following equation

$$r_b = max\left[\beta\frac{\sum_{i=1}^{n_c} r_i}{n_c}, \max(r_i)\right] \tag{3.2}$$

in which n_c is the number of pore throats connected to the same pore body, and β is an aspect ratio whose value is set between zero and one in this work. The pore shape is determined by a shape factor G that relates the area of the pore body or throat to its perimeter through the following relationship.

$$G = \frac{A}{P^2} \tag{3.3}$$

In this study the shape factor varied between 0.01 and 0.04811 for pore bodies and pore throats, which means the pore geometry ranged from silt-shaped to equilateral triangles, as shown in Fig. 3.1.

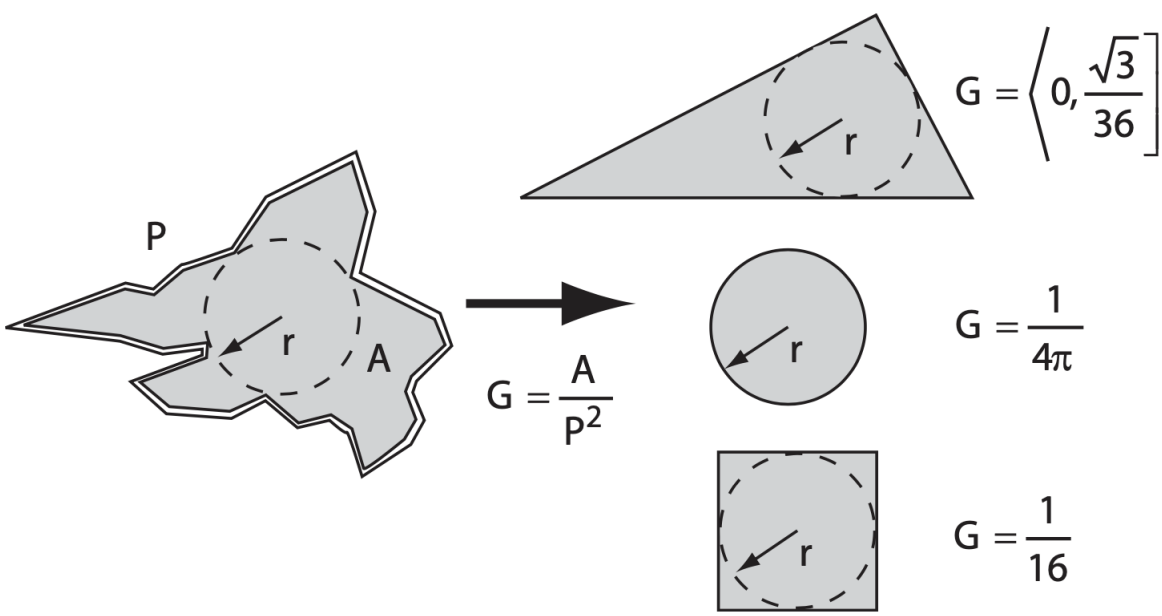

Figure 3.1: *Explanation of the difference in shape factors (after Valvatne (2004)).*

In addition to the pore-throat radius, the distributions of its length and pore shape follow

the same truncated Weibull distribution with the same parameters δ and γ. Unlike the pore-throat radius and pore shape, the minimum pore-throat length l_{min} and its maximum value l_{max} were adjusted from one pore network to another such that all of them maintained a porosity between 0.09 and 0.31. More specifically, for pore networks in which the pore-throat radius varied between 0.1 and 10 μm, l_{min} and l_{max} were selected randomly between 1 and 100 μm, so as to have a porosity in the aforementioned values. The same approach was also used for pore-throat radius distributions between 1 and 100 μm with the minimum and maximum pore-throat lengths being randomly selected between 10 and 1000 μm. The Weibull distribution parameters, δ and γ for the pore-throat lengths were equal to the parameters of the pore-throat radius distribution for each network. This was done across our entire data set, regardless of parameters listed in Table 3.1.

Table 3.1: *Finalized network parameters for database generation, totaling up to 240 generated networks.*

Pore Size Distribution	Pore Shape Distribution	Pore Throat Radii	Clay Content	Coordination Number	Contact Angle
δ: 0.2 γ: 24	δ: 0.2 γ: 24	Min: 0.1 Max: 10	0%	2	0°
δ: 0.2 γ: 12	δ: 0.2 γ: 1.35	Min: 1 Max: 100	20%	4	60°
δ: 0.2 γ: 6				6	
δ: 0.2 γ: 3					
δ: 0.2 γ: 1.35					

By altering the pore-scale properties summarized in Table 3.1, we generated 240 different types of pore networks that cover a broad range of petrophysical properties, which were subsequently used for rock typing.

3.1.2 Sandstones

In addition to the 240 synthetic networks, we constructed six additional networks based on a Mt. Simon sandstone sample from Kohanpur et al. (2020), a Berea sandstone from Valvatne (2004), and four Fontainebleau sandstone samples reported by Lindquist et al. (2000) and

Arns et al. (2003). For Mt. Simon and Berea sandstone we used the original pore networks provided by Kohanpur et al. (2020) and Valvatne (2004), that had been extracted from their digital images. The four pore networks for Fontainebleau sandstone were generated with pore-body and pore-throat size distributions, coordination numbers, and water relative permeabilities that matched those reported by Lindquist et al. (2000) and Arns et al. (2003).

3.1.3 Flow simulation

To determine the representative elementary volume (REV) for each network, we first carried out simulation of single-phase flow and conductivity to compute the formation factor and permeability for pore-network sizes 10, 20, 30, 40, 50, 60, and 65, each with 10 realizations. Recall that the network size represents the number of pore bodies along each of its sides. After determining the REV, we carried out simulation of oil flooding in the pore networks, initially saturated by water, and computed the water relative permeability for a network of size REV, using 100 realizations. To determine the average relative permeability for each network, we first interpolated all their curves using the Makima interpolation method in MATLAB. This led to evenly-spaced water saturations from $S_w = 1$ to $S_w = S_{wc}$, the critical water saturation at which water relative permeability vanishes. We then averaged over the interpolated curves to determine the representative k_{rw} curve for each network (Ahmed, 2001). We also averaged the absolute permeability, porosity, formation factor, and the mode of the pore-throat radii across all the 100 realizations for each network.

3.2 Two-phase rock typing

Recently, Ghanbarian et al. (2019) proposed a rock typing method based on data for the formation factor and the absolute permeability measurements. In their approach rocks with similar characteristic pore sizes are classified into the same type. In this section, we apply concepts from critical-path analysis to develop a new rock typing approach based on two-phase flow data and water relative permeability k_{rw}. Similar to the rock typing approach

of Ghanbarian et al. (2019), we classify rocks that have similar critical pore radius r_c at the same effective water saturation S_e. Thus, we should convert the $S_w - k_{rw}$ curves into $S_e - r_c$ ones. For this purpose, we apply critical-path analysis that has been successfully utilized to model water relative permeability in porous media (Ghanbarian and Hunt, 2017; Ghanbarian et al., 2016a; Ghanbarian-Alavijeh and Hunt, 2012; Hunt, 2001). We use the following equation that invokes a power-law relationship between k_{rw} and r_c from critical-path analysis (Ghanbarian, 2020; Hunt, 2001).

$$k_{rw} = \left[\frac{r_c(S_w)}{r_c(S_w = 1)}\right]^{\alpha} \tag{3.4}$$

where $r_c(S_w)$ and $r_c(S_w = 1)$ are critical pore-throat radius under partially- and fully-saturated conditions, respectively. In Eq. (3.4), $\alpha = 3$, if pore-throat length is linearly proportional to its radius, and $\alpha = 4$, if it is independent of its radius (Ghanbarian et al., 2016c).

To estimate α one may apply the critical-path analysis that links the permeability to the formation factor and critical pore-throat radius by the following equation (Katz and Thompson, 1986):

$$k = \frac{r_c^2(S_w = 1)}{CF} \tag{3.5}$$

where C is a constant whose value is $72.2/4 \simeq 18.05$ when the pore-throat length is linearly proportional to its radius, or $\alpha = 3$, and $53.5/4 \simeq 13.375$ when pore-throat radius is independent of its radius, or $\alpha = 4$ (Ghanbarian et al. (2016c)). The value of $r_c(S_w = 1)$ is determined from the mode of the pore-throat size distribution (Katz and Thompson, 1986). If the permeability, formation factor and critical pore-throat radius are available, one may determine which C value results in more accurate estimates of the permeability by Eq. (3.5). After α is determined, one calculates the ratio $r_c(S_w)/r_c(S_w = 1)$ via Eq. (3.4).

Once the critical pore radii at various water saturations are determined, one may calculate an effective water saturation defined by

$$S_e = \frac{S_w - S_{wc}}{1 - S_{wc}} \tag{3.6}$$

Where S_w is water saturation, and S_{wc} is its critical value at which k_{rw} vanishess. The value of S_{wc} is determined from the $S_w - k_{rw}$ curve. To identify different rock types, $r_c(S_w)/r_c(S_w = 1)$ should be plotted against S_e. Those curves that collapse together are then considered as belonging to the same rock type. The proposed rock typing method based on the data for two-phase flow and its steps are summarized in Fig. 3.2.

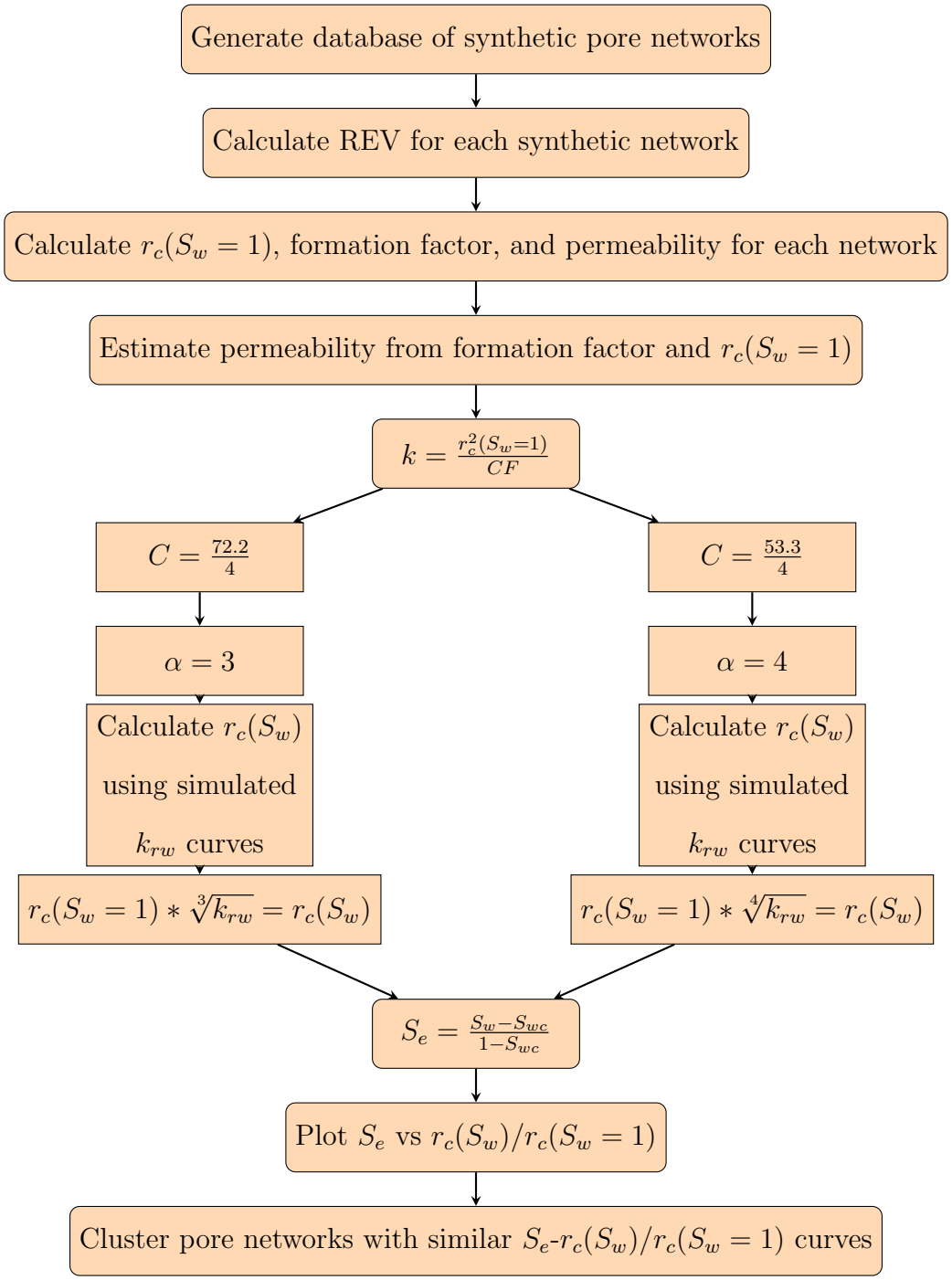

Figure 3.2: *Outline of the steps followed in our rock typing process.*

3.3 Clustering Method

After converting the $S_w - k_{rw}$ curves to $S_e - r_c(S_w)/r_c(S_w = 1)$ ones, we apply the curve clustering approach using the open-source toolbox developed by Gaffney (2004), available at http://www.datalab.uci.edu/software/CCT/. The clustering toolbox uses two regression mixing models, namely, polynomial regression (lrm) and spline regression (srm), in order to insert similar curves in the same cluster. The curve clustering method uses a regression mixture equation with up to four transformations, given by

$$y = c[ax + b]B + d + e \tag{3.7}$$

in which the values within the square brackets represent the transformed regression matrix, c is related to the scaling in the measurement space, d is linked to translation in the measurement space, a is related to the scaling in time, and b is related to translation in time. In the present study we assume that S_e is the time, while $r_c(S_w)/r_c(S_w = 1)$ is the measurement space, or S_e and $r_c(S_w)/r_c(S_w = 1)$ are represented by the horizontal and vertical axes of a plot. Using these transformation parameters, we attempted to cluster all the data into representative rock types as discussed earlier.

Chapter 4

Results and Discussion

Results from this study indicate that our two-phase rock typing successfully clustered S_e-$r_c(S_w)/r_c(S_w = 1)$ into representative rock types; with all 240 synthetic samples and six sandstone samples being clustered into twelve representative rock types. Our method also highlights the importance of incorporating two-phase data into rock typing procedures as our results indicated a different number of rock types when compared to commonly used single-phase rock typing methods. The complete results from both our pore network simulations and our rock typing are discussed below.

4.1 Pore Network Modeling Results

Before the final data set could be created, and each network iterated over one hundred iterations, we first had to determine REV for each network. For this purpose, we found that REV in our networks ranged between a lattice size of 60 and 65, with ten randomly selected networks and their corresponding REV plots shown in Fig. 4.1. As displayed in Fig. 4.1(b) formation factor and permeability remained constant independent of network size at high lattice sizes. Additionally, as displayed in Fig. 4.1(c) at these larger lattice sizes, scatter in relative permeability data disappears, indicating that REV has been met. For simplicity sake, we chose a network size of 65 for every network, and iterated each network

one hundred times. Following the averaging steps discussed in Chapter 3 we acquired the following results.

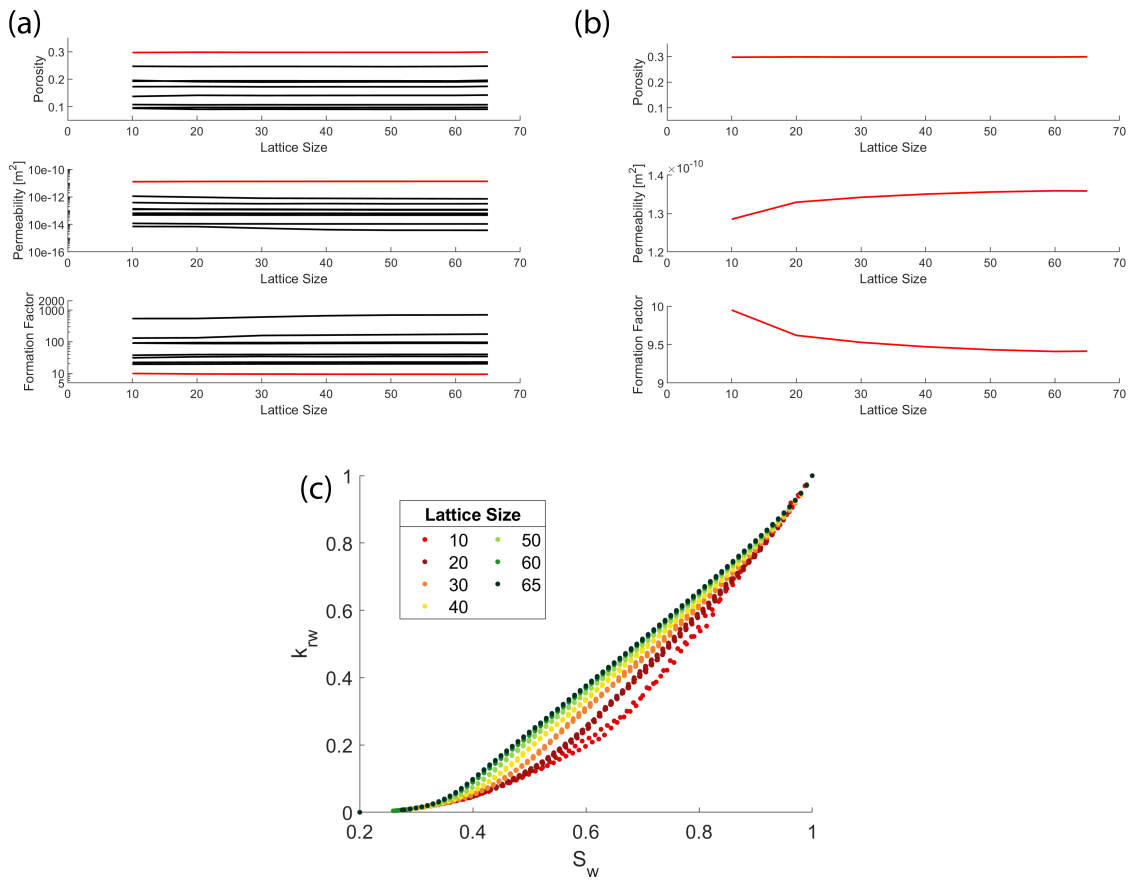

Figure 4.1: *Plot (a) shows REV plots for ten randomly selected synthetic networks. With plot (b) showing single-phase REV for the network highlighted in red in plot (a). Plot (c) displays S_w-k_{rw} curves across lattice sizes for the same red network in plot (a). Note, that variations in data completely disappear by a lattice size of 65.*

As shown in figure 4.2, our data set covers a wide range of permeability, formation factor, and porosity values. With the minimum, maximum, average, and median values for each of these listed in Table 4.1. Our permeability values span six orders of magnitude while formation factor spans almost four orders of magnitude.

Table 4.1: *Overview of entire data set, indicating the minimum, maximum, average, and median values for permeability, formation factor, and porosity.*

	Permeability [m^2]	Formation Factor	Porosity	$r_c(S_w = 1)$ [m]
Maximum	1.359e-10	1501.16	0.3132	9.1090e-05
Minimum	1.680e-16	5.709	0.0747	1.7396e-06
Average	1.515e-11	127.146	0.1752	3.3857e-05
Median	1.01e-12	30	0.1657	1.7403e-05

After plotting data as shown in Fig. 4.2, we then plotted the data from Fig. 4.2(a) into the plot that Philip Nelson proposed in Nelson (1994). We found that the majority of our data set falls into the consolidated and unconsolidated sands (Fig. 4.3). This is in accord with the basis of our pore network modeling package we employed (Valvatne, 2004) as it was designed to model unimodal sandstones.

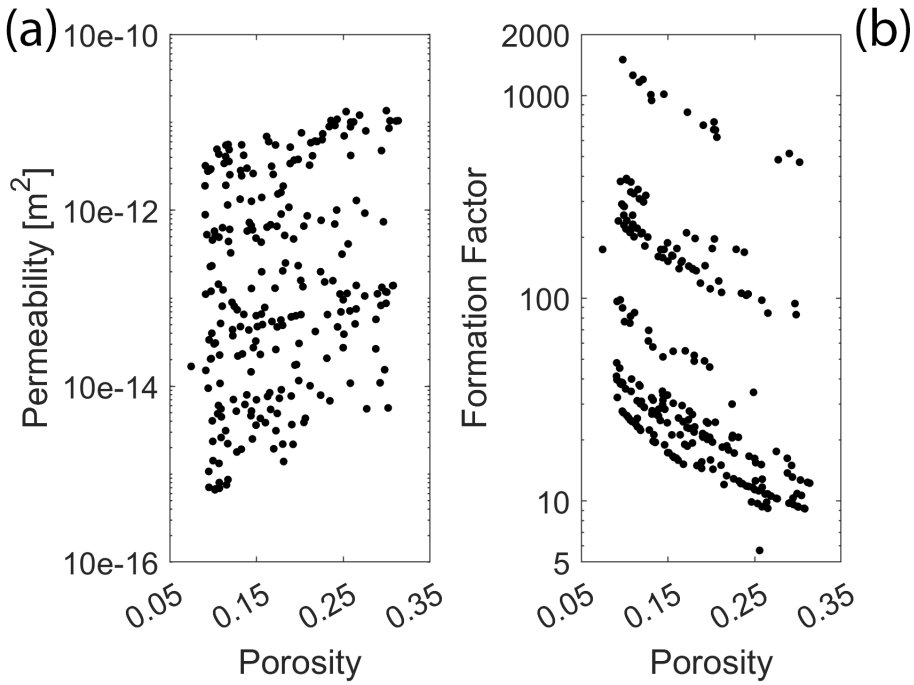

Figure 4.2: *Single-phase results from our synthetic database and matched real would results. Plot (a) shows the porosity versus permeability for each sample while plot (b) shows porosity versus formation factor for each sample.*

Further analysis of our data set finds that our pore networks match data in the literature, with our range of porosity, permeability and formation factor values matching experimental data (Bashtani et al., 2016; Bourbie and Zinszner, 1985; Byrnes et al., 2008; Ghanbarian et al., 2019). For example, Bourbie and Zinszner (1985) collected porosity and permeability data from Fontainebleau sandstone samples and found that porosity and permeability ranged between 0.02 and 0.30 and between 10^{-16} and 10^{-11} m^2, respectively, similar to the range of our data, which supports the validity of our pore networks. We also found that both Bashtani et al. (2016) and Byrnes et al. (2008) measured formation factor and permeability on 2200 sandstone samples from the western US basins and found values that agree well with our data, further supporting the validity of our pore networks.

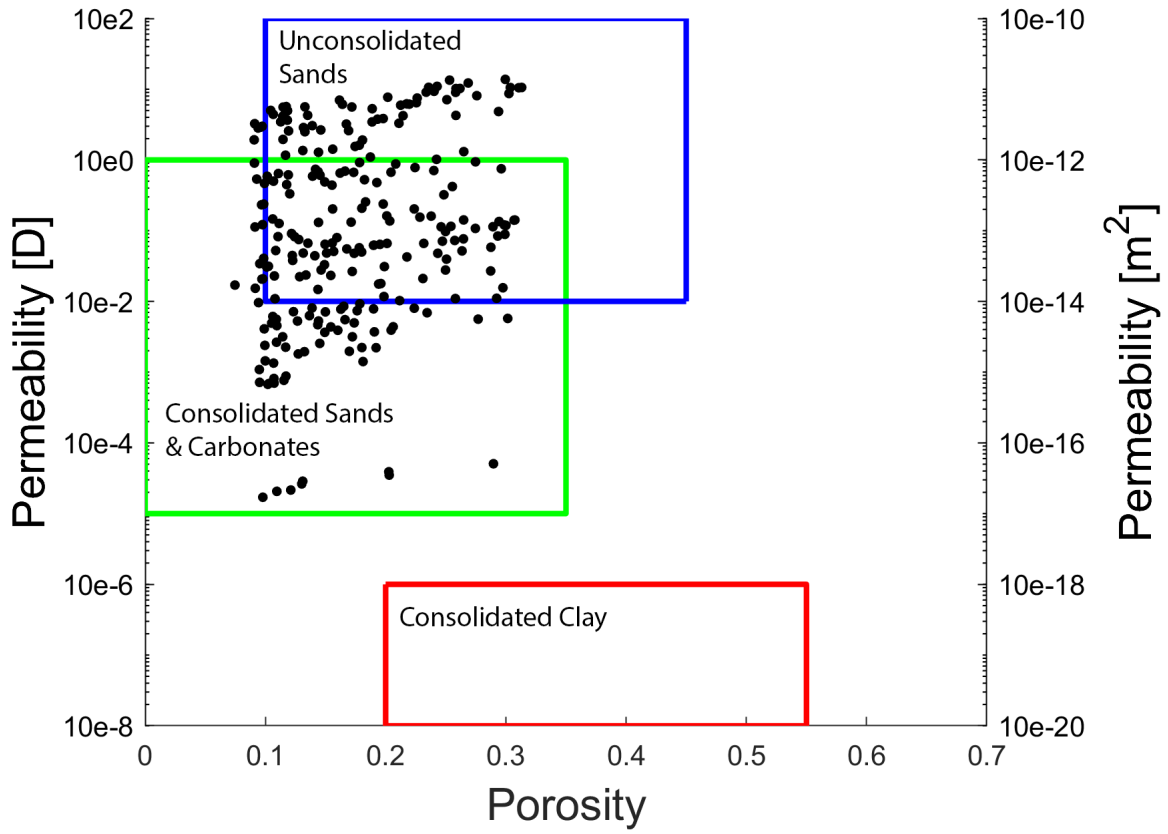

Figure 4.3: *Comparison of simulated data to different types of rocks following theory from Nelson (1994). The red box represents the area in which consolidated clays will fall. With the green and blue boxes representing the range in which consolidated sands and carbonates or unconsolidated sands will respectively fall. The majority of our data set is classified as consolidated or unconsolidated sands.*

Comparing our Fontainebleau sandstone networks to those described by Lindquist et al. (2000) and Arns et al. (2003) results in the following Table 4.2. As can be seen, our simulations match those in the literature relatively well, with identical porosity values, slightly higher permeability values, and almost identical critical water saturations.

Table 4.2: *Results form pore network models of Fontainebleau sandstones created using the pore-network models from Valvatne (2004), and compared to experimental results obtained in Lindquist et al. (2000) and Arns et al. (2003).*

Created Networks			Arns et al. (2003) Networks		
Porosity	Permeability $[m^2]$	S_{wc}	Porosity	Permeability $[m^2]$	S_{wc}
0.074	1.68e-13	0.68	0.075	1.09e-13 - 1.67e-13	0.45 - 0.58
0.127	7.40e-13	0.40	0.13	4.56e-13 - 7.16e-13	0.34 - 0.39
0.144	1.29e-13	0.30	0.15	6.04e-13 - 1.12e-12	0.31 - 0.37
0.223	2.00e-12	0.25	0.22	1.77e-12 - 3.05e12	0.28 - 0.34

Comparing the simulated water relative permeability curves in our study with those from Arns et al. (2003) showed nearly identical critical water saturations and good agreement over the entire range of saturation. It should be noted, however, that the Arns et al. (2003) $S_w - k_{rw}$ curves show considerable spread, with the curves, denoted by black dots, reported in Fig. 4.4 being the reported average $S_w - k_{rw}$ curves from Arns et al. (2003).

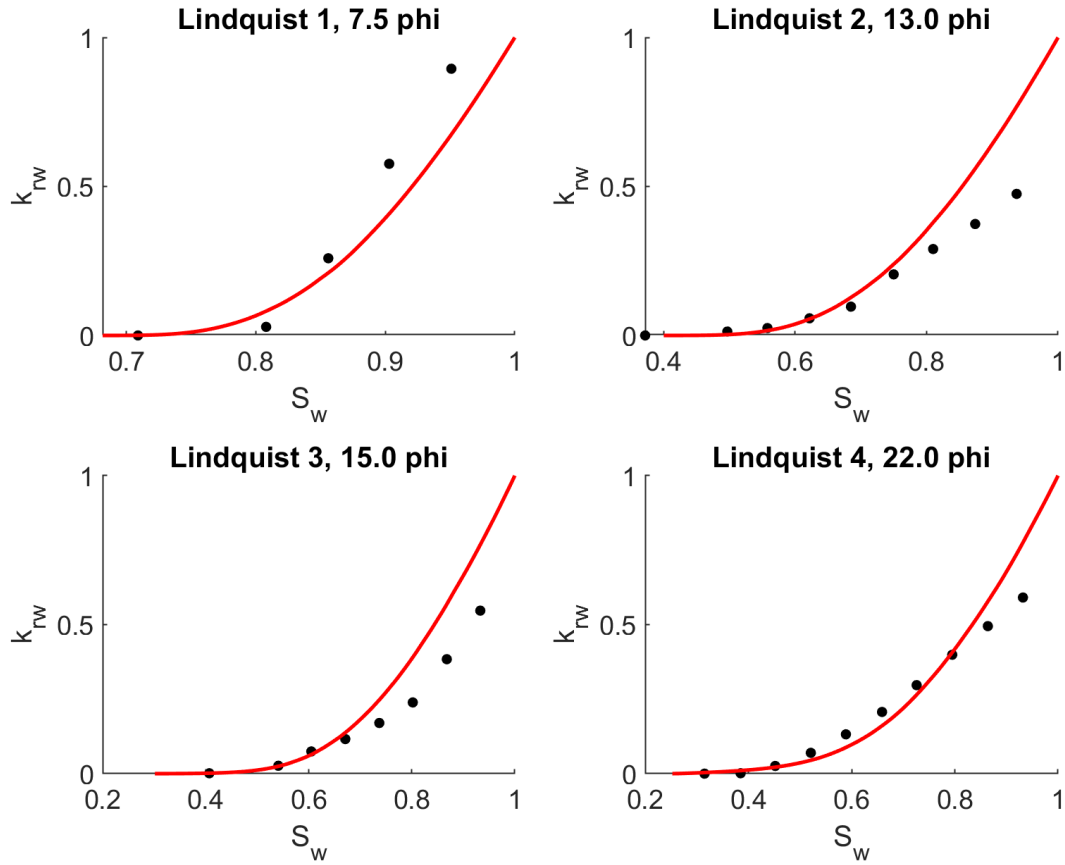

Figure 4.4: *Match between our created networks and the four Fontainebleau sandstones depicted in Lindquist et al. (2000) and Arns et al. (2003). Black dots mark the literature data, with red lines denoting our match.*

4.2 The Porosity - Formation Factor Relationship

Although the $\phi - k$ plot in Fig. 4.2(a) seems scattered, the plot of the data for $\phi - F$ relation follows specific trends (Fig. 4.2(b)). Further analysis of the $\phi - F$ data indicated that there exist eight distinct groups of data, with two outliers denoted as pink circles (Fig. 4.5). Using modified Archie's law, Eq. (2.1), we calculated the cementation exponent for each of the eight clusters, and found that constant a and the exponent m varied, respectively, from 2 to 134 and 0.81 to 1.22 (Table 4.3). Such values of m closely match those reported by Porter and

Carothers (1971) from wells located in offshore Californian Pliocene sediments and offshore Texas-Louisiana Miocene sediments. Alreshedan and Kantzas (2016) also used the same computer program developed by Valvatne (2004) and reported m values between 1.21 and 2.24, albeit with a pre-set value $a = 1$. The values determined by Porter and Carothers (1971) were from *in-situ* measurements of well logs of fully water-saturated formations. These values are also consistent with those suggested by Dashtian et al. (2015). They analyzed extensive resistivity well logs to highlight the effect of long-range correlation and multifractality of the data on value of the the Archie's law exponent m at reservoir scales. Dashtian et al. (2015) stated that, "Only when the resistivity logs are analyzed over several hundred meters do long-rage correlations manifest themselves."

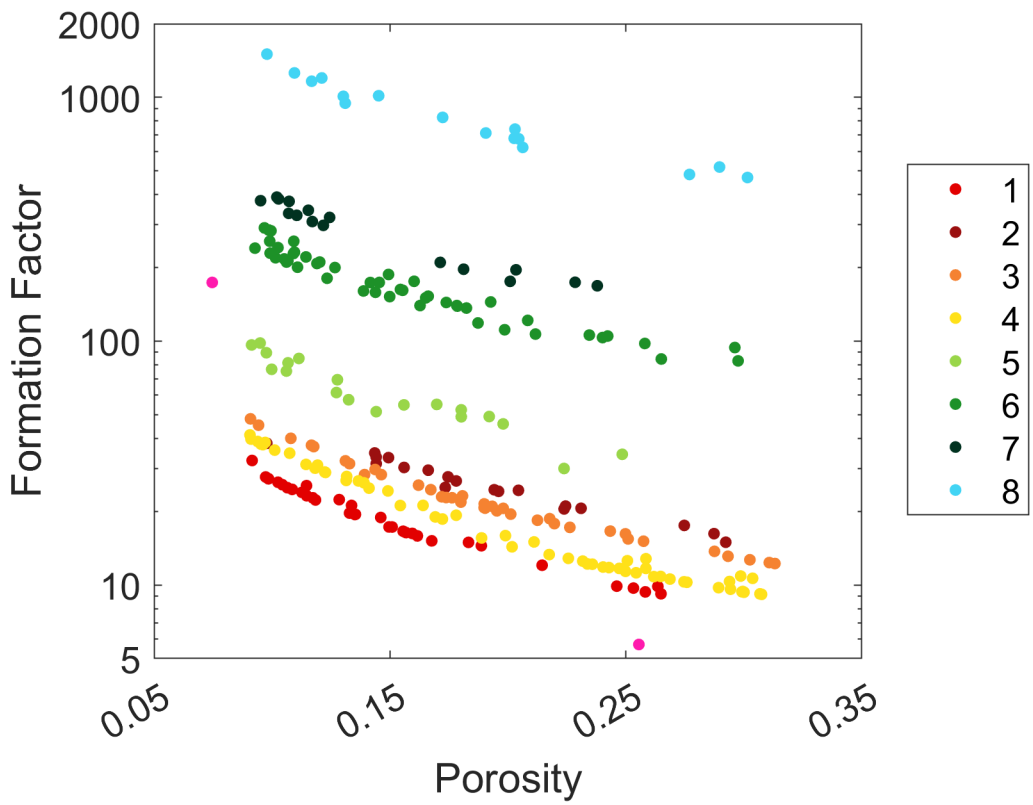

Figure 4.5: *Individual clusters that were used to calculate the cementation exponent for Archie's law. The values calculates using Archie's Law for each cluster are given in Table 4.3*

For each cluster of the $\phi - F$ data shown in Fig. 4.5, we also calculated the exponent m using modified Archie's law when $a = 1$, Eq. (2.1), and found $1.47 < m < 5.13$ (see Table 4.3), which resemble more closely the values reported by Müller-Huber et al. (2015), as well as others in the literature, with the average value of m being 2.18 and a median of 1.90 (Archie, 1950; Martin et al., 1996; Müller-Huber et al., 2015; Sen et al., 1988).

Table 4.3: *Values derived using Archies Law for each of the groups outlined in Figure 4.5*

Cluster	a	m	r^2	Cluster	a	m	r^2
1	1	1.47 - 1.72	-	1	2.17	1.10	0.99
2	1	1.57 - 2.24	-	2	6.53	0.81	0.91
3	1	1.61 - 2.16	-	3	3.37	1.11	0.99
4	1	1.53 - 1.99	-	4	2.23	1.22	0.99
5	1	1.88 - 2.54	-	5	8.13	1.02	0.93
6	1	2.30 - 3.73	-	6	24.42	1.01	0.93
7	1	2.52 - 3.57	-	7	38.11	1.00	0.97
8	1	3.14 - 5.13	-	8	134.40	1.02	0.97

4.3 Two-phase Rock Typing

The simulated water relative permeabilities for all 246 samples are shown in natural and logarithmic scales in Figures 4.6(a) and 4.6(b). Figure 4.6(a) indicates that the results cover a wide range of $S_w - k_{rw}$ curves, while according to Fig. 4.6(b) the critical water saturation varies between 0 and 0.29 in the synthetic pore networks, and between 0.012 and 0.68 in the actual sandstone samples. The calculated $S_e - r_c(S_w)/r_c(S_w = 1)$ curves are also shown in Figures 4.6(c) and 4.6(d).

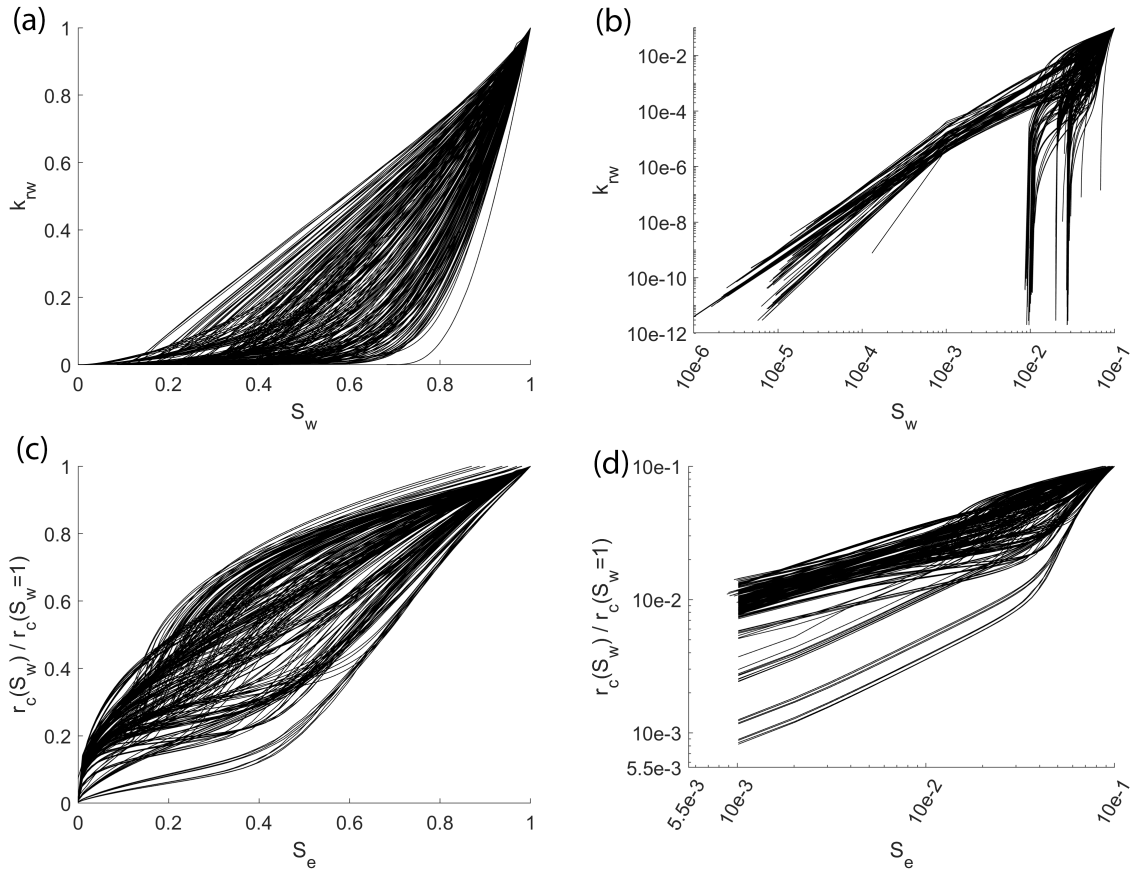

Figure 4.6: *Complete overview of S_w-k_{rw} and S_e-$r_c(S_w)/r_c(S_w = 1)$ curves for all 246 samples in both natural and log-log scales.*

The results for clustering the S_e-$r_c(S_w)/r_c(S_w = 1)$ curves are presented in Fig. 4.7. We tested all the available clustering methods listed in Table 4.4, and found, however, that the only clustering method that produces reliable results is a polynomial regression mixing model with a transformation of the form $[x+b]B$. All other transformations produce errors, or do not group the rock samples appropriately (results not shown). We used the $[x+b]B$ form and evaluated the transformation with polynomial whose order ranged from one to five, as well as varying the number of clusters from seven to twelve. Table 4.4 summarizes the clustering parameters used in our study.

Table 4.4: *Overview of all the curve clustering methods that were attempted to calculate representative rock types.*

Regression Model	Transformation parameters	Polynomial Order	Number of Clusters
lrm	$[ax+b]B$	1	7
srm	$[x]B+d$	2	8
	$[x+b]B$	3	9
	$[x+b]B+d$	4	10
	$c[x]B+d$	5	11
	$[ax+b]B+d$		12
	$c[x+b]B+d$		
	$c[ax+b]B+d$		

We also assessed multiple clustering methods using various cluster numbers and polynomial orders listed in Table 4.4 and found that a polynomial order of three and twelve clusters can appropriately group the data into distinct clusters. A smaller cluster number caused samples with unlike critical pore radii to group together. For polynomial order less than three, it was found that one does not always obtain the same reliable results by applying the clustering method, whereas a polynomial order of three provided the most consistent results out of all the polynomial orders tried. The results of our clustering method using a cluster number of twelve and polynomial order of three are shown in Fig. 4.7. Samples with similar normalized critical pore-throat radii, $r_c(S_w)/r_c(S_w = 1)$, across the range of the effective water saturation S_e were clustered together into similar rock types.

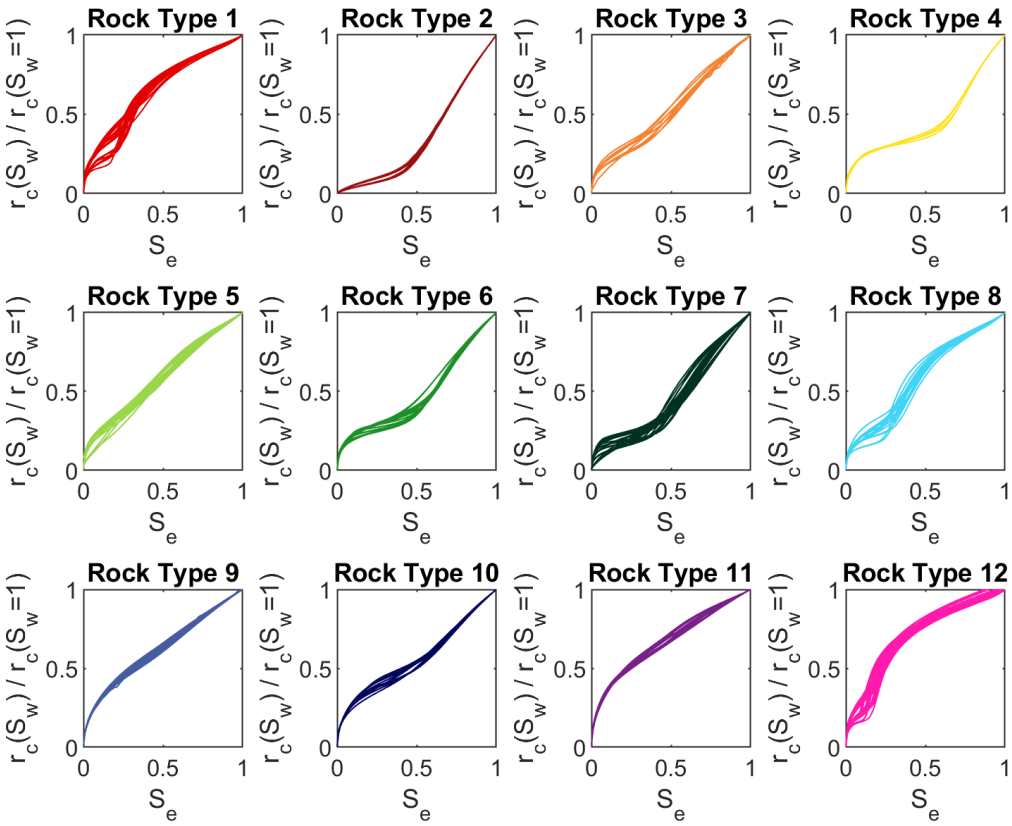

Figure 4.7: *All twelve rock types separated by color and plot. These clusters include all 246 S_e-$r_c(S_w)/r_c(S_w = 1)$ curves. Clusters were determined using a transformation parameter of $[x + b]B$, polynomial order of three, and a cluster number of twelve.*

As mentioned earlier, Mirzaei-Paiaman et al. (2019b) proposed recently another method of rock typing based on two-phase flow data. They used experimental data for two-phase flow in rock samples from the Bangestan group and Asmari formation in Iran, and defined a "true" effective mobility function, the aforementioned TEM function, previously discussed in Sec. 2.2, which is determined from permeability measured at various saturations, porosity, and fluid viscosity in order to cluster rocks based on their ability to permit fluid flow (Faramarzi-Palangar and Mirzaei-Paiaman, 2020a; Mirzaei-Paiaman and Ghanbarian, 2020; Mirzaei-Paiaman et al., 2019b). In their methodology rocks with similar fluid flow characteristics have similar TEM functions, and rocks with greater TEM values are considered to be of

higher quality reservoirs. Using the two experimental data sets from Iran, Mirzaei-Paiaman et al. (2019b) showed that using only relative permeabilities to classify rock samples into representative types may lead to their inaccurate classification. They also demonstrated that rocks with similar TEM curves may not necessarily have similar relative permeabilities, recall Fig. 2.3. These findings are in accord with our own findings in the present study. We also find that there might be significant scatter in the water relative permeability data in each individual rock type; see Fig. 4.8 that indicates that the water relative permeability curves are scattered within each cluster, particularly at the dry end near the critical water saturation where the water relative permeability vanishes. Our results demonstrate the profound impact of converting the $S_w - k_{rw}$ curves to the $S_e - r_c(S_w)/r_c(S_w = 1)$ ones.

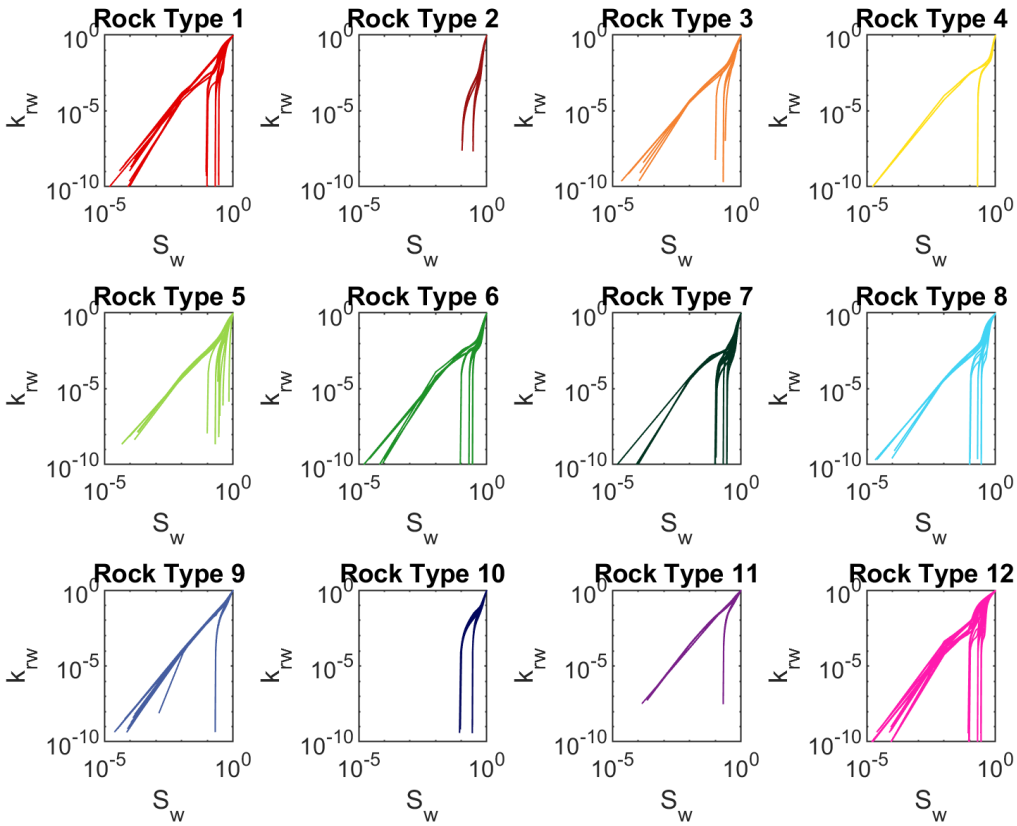

Figure 4.8: *Individually clustered S_w-k_{rw} curves, that have been grouped into their representative rock types. Note, the scatter towards the dry end of the S_w-k_{rw} curves.*

When comparing the rock types determined from our proposed rock typing method to the method outlined in Mirzaei-Paiaman et al. (2019b), we find that they do not correlate (Fig. 4.9), with our picked rock types inaccurately grouping alike TEM curves. However, this result is not surprising, given that our proposed method and the method proposed by Mirzaei-Paiaman et al. (2019b) use two different theoretical backgrounds to determine rock types. While our method is based on critical path analysis and critical pore sizes, the method from Mirzaei-Paiaman et al. (2019b) is based on Darcy's law, and the ability for a sample to permit fluid flow under applied pressure gradients . Additionally, different parameters are used to classify rock types, with TEM curves calculated using relative permeability curves, fluid viscosity, and porosity (Eq. 2.10), and our method is calculated using critical pore sizes and wetting phase relative permeability curves.

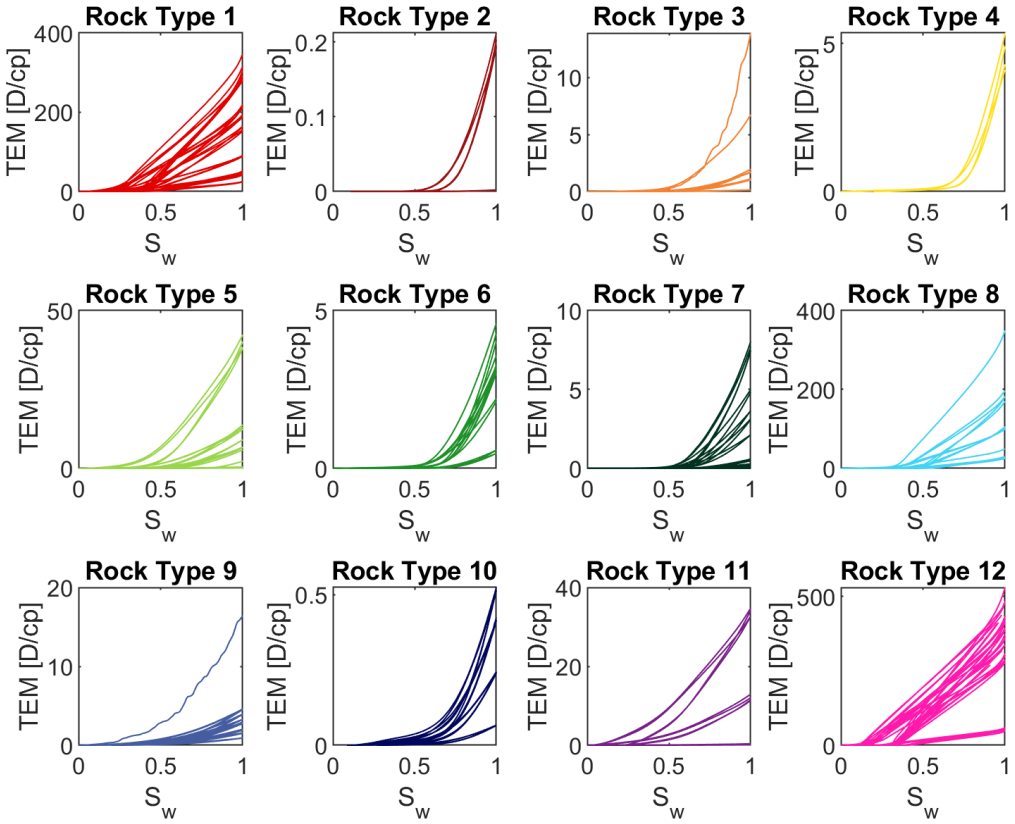

Figure 4.9: *Comparison of our picked rock types with the TEM curves. As shown, our method results in different rock types when compared to the TEM function.*

Due to these differences, we employed the use of our clustering method to classify rock types based on their TEM curves. In doing this, we tested all clustering parameters outlined in Table 4.4 and found the most success, or rather only success, using a spline regression mixing model with a transformation of $[x]B + d$, a polynomial order of four, and cluster number of eight. These clustering parameters produced the results shown in Fig. 4.10, with eight rock types being defined. All other clustering parameters either gave no results, or grouped all samples into one rock type. As demonstrated however, our clustering method does a suboptimal job of clustering TEM curves, grouping unlike TEM curves into similar rock types.

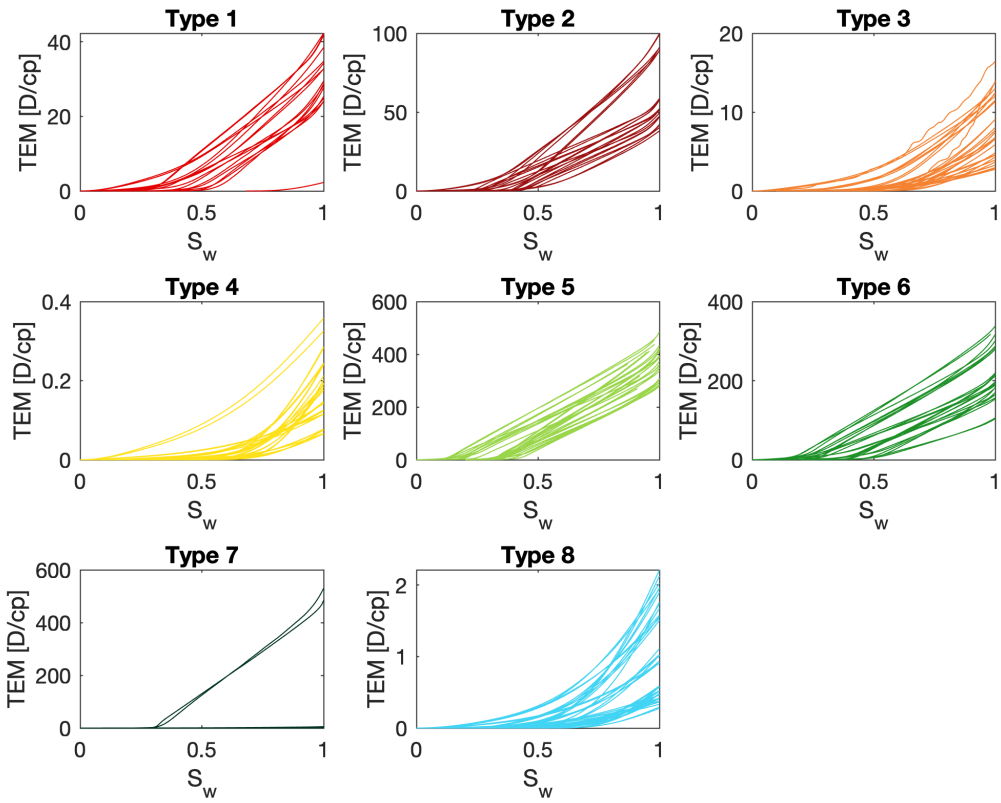

Figure 4.10: *Individually clustered $S_w - TEM$ curves using the clustering package from Gaffney (2004). Using our curve clustering method and TEM curves, there are eight rock types determined. However, our curve clustering methods does not define rock types as accurately using TEM curves when compared with $S_e - r_c(S_w)/r_c(S_w = 1)$ ones.*

4.4 Comparison with single-phase rock typing

Next, we compare the results presented so far with the recently developed method for rock typing based on single-phase flow data, proposed by Ghanbarian et al. (2019). In their method, the permeability is plotted against the inverse of formation factor, and rocks with similar characteristic pore sizes are classified into the same rock type. They analyzed a total of 275 experimental samples, including 48 from Katz and Thompson (1986), 107 from Sen et al. (1988), and 120 from Sen et al. (1990). In their method the formation factor is related

to the permeability by the following equation, originally suggested by Johnson et al. (1986)

$$k = \frac{\Lambda^2}{8F} \tag{4.1}$$

where Λ is a characteristic size for dynamically connected pores. Using Eq. (4.1), Ghanbarian et al. (2019) grouped samples with similar Λ values into the same rock type. Following them, we plotted the permeability k against $1/F$ for the 246 samples studied here, identified 15 rock types; Fig. 4.11, and calculated Λ, the results of which are reported in Table 4.5.

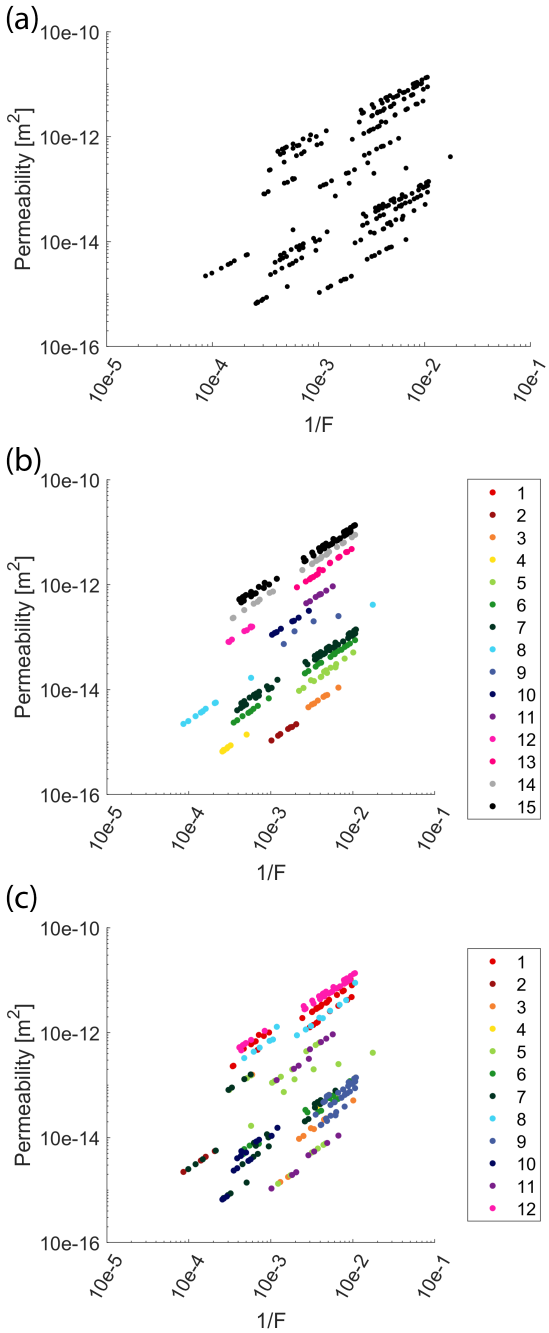

Figure 4.11: *Comparison between the rock typing method outlined in Ghanbarian et al. (2019) and our proposed rock typing method. With plot (a) showing unclustered data, plot (b) showing the clustered data using Eq. 4.1, and plot (c) showing the clusters determined using our proposed rock typing method.*

We show in Fig. 4.11(a) the plot of the permeability versus the inverse of formation factor,

using the data obtained by pore-network simulations. The data display linear patterns with positive slope, indicating that the permeability decreases as formation factor increases, in accord with the critical-path analysis and Eq. (3.5). The results for rock typing based on single-phase data and the identified clusters using the method proposed by Ghanbarian et al. (2019) are presented in Fig. 4.11(b), which indicate that there exist fifteen main rock types based on single-phase data. For comparison, we also show the results for rock typing based on two-phase flow data in Fig. 4.11(c). Displaying the twelve rock types identified based on the two-phase data analysis on a $1/F - k$ plot demonstrates that samples within the same type identified by two-phase flow data may not necessarily belong to the same cluster based upon rock typing with single-phase data. These results highlight the importance of incorporating two-phase data for rock typing, particularly in mix-wet reservoirs.

Table 4.5: *Values derived using Eq. 4.1 for our dataset. See Fig. 4.11(b) for cluster number.*

Cluster number	Λ [μm]	Cluster number	Λ [μm]
1	1.45	9	15.76
2	2.96	10	29.80
3	3.66	11	36.33
4	4.84	12	47.45
5	6.46	13	65.54
6	8.18	14	82.19
7	9.86	15	98.97
8	13.84		

Ghanbarian et al. (2019) reported a range of Λ between 0.05 and 30 μm. In their study, the permeability spanned nearly eight orders of magnitude, from 10^{-16} m^2 (10^{-1} mD) to 10^{-8} m^2 (10^7 mD). In the present study, the permeability varies between 10^{-16} m^2 (10^{-1} mD) to 10^{-10} m^2 (10^5 mD), six orders of magnitude variations. Large Λ in our study

correlate well with the distribution of pore-throat radius in the simulations. For instance, cluster 15, Fig. 4.11(b) with $\Lambda = 98.97$ μm encompassing all the pore networks in our simulations that have a Weibull distribution with the parameters $\delta = 0.2$ and $\gamma = 24$, or $\gamma = 12$, as well as $r_{\min} = 1$ μm and $r_{\max} = 100$ μm. Recall that in Eq. (3.1) δ and γ control the shape of the pore-throat radius distribution, and larger γ correspond to narrower distributions. Rock types that display smaller Λ also exhibit broader pore-throat radius distributions corresponding to $\gamma = 1.35$ and $\gamma = 3$, Eq. (3.1), and were constructed with smaller pore-throat radii varying between 0.1 and 10 μm.

4.5 Effect of Network Parameters

4.5.1 Contact Angle

In the literature, the effect of contact angle on the relative permeability has been investigated extensively (Anderson, 1987; Blunt, 1997; Dicarlo et al., 2000; Li et al., 2005; Mahmud et al., 2007; Xu et al., 2014). In the present study, we used two initial contact angles, i.e., 0 and 60°. As may be expected, the results of pore network simulations with contact angle of 0° are not comparable with those for networks with a contact angle of 60° since, as stated earlier, those networks do not necessarily have the same pore-throat length characteristics, i.e., randomly-selected minimum and maximum pore-throat radii. Nonetheless, in Figures 4.12(a) and (b) we show the results for the two contact angles. They indicate that the differences between the plots for $S_w - k_{rw}$ with contact angles 0 and 60° are not substantial. This is consistent with the results of Li et al. (2005), Hao and Cheng (2010), and Landry et al. (2014) who demonstrated that the wetting-phase relative permeability does not significantly vary from nearly perfectly-wet to neutrally-wet conditions.

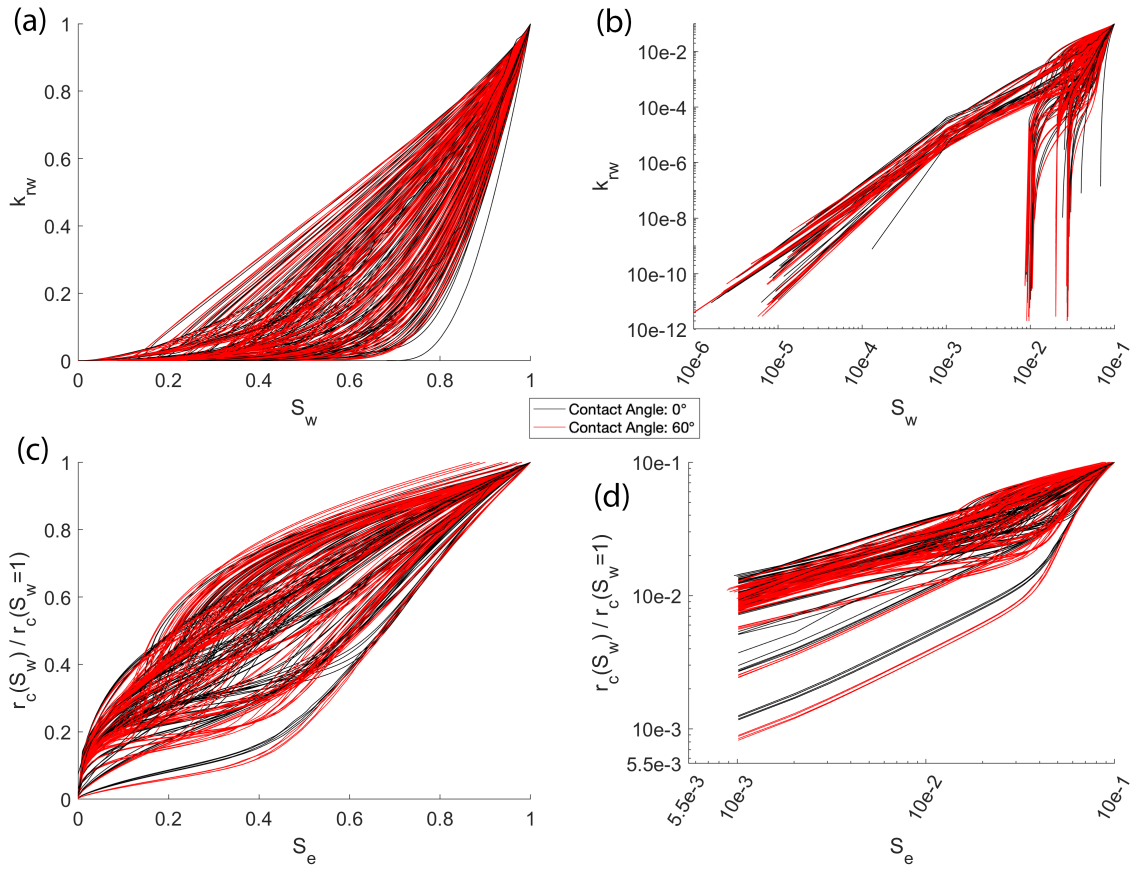

Figure 4.12: *Complete data set broken down by initial contact angle, with plots (a) and (b), the S_w-k_{rw} curves, and plots (c) and (d) being the S_{wc}-$r_c(S_w = 1)/r_c(S_w = 1)$ curves.*

Because the value of contact angle affects water relative permeability, our rock typing method indirectly incorporates its influence through the $S_w - k_{rw}$ curves and converting them to $S_{wc} - r_c(S_w = 1)/r_c(S_w = 1)$ plots for grouping rocks. We find that all but one rock type, rock type four, included networks that have both contact angles of 0 and 60° (Fig. 4.13). This clearly indicates that our rock typing method classifies rocks based on various petrophysical properties, and does not strongly rely on a single property, such as the permeability, formation factor, or contact angle.

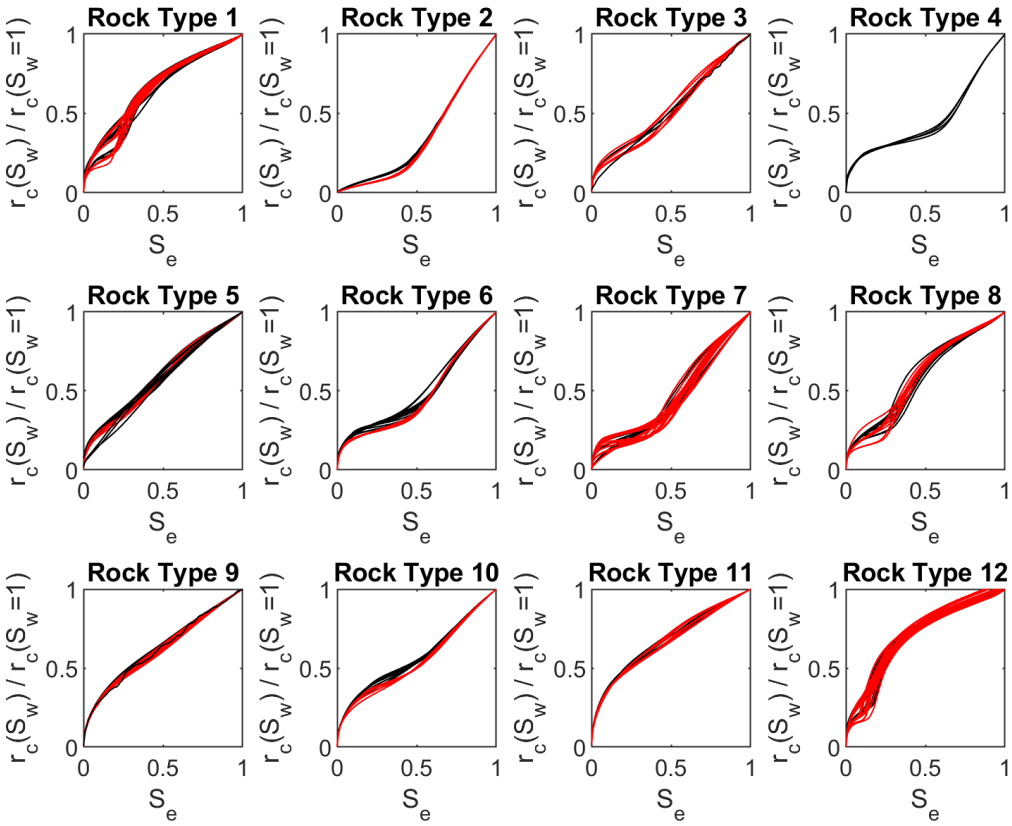

Figure 4.13: *Illustrates the influence of contact angle on our rock typing process. With all but one rock type containing samples with contact angles of 0°, represented by black curves, and 60°, represented by red curves.*

4.5.2 Pore-throat Size

To investigate the effect of pore-throat radius and its range on the proposed rock typing, we further analyzed our data, particularly those for the 240 synthetic samples. The results from two ranges of pore-throat sizes, i.e., $0.1 - 10$ μm and $1 - 100$ μm are shown in Fig. 4.14. As expected, the pore networks with 1 μm $\leq r \leq$ 100 μm generally display larger critical pore-throat radii compared to those with 0.1 μm $\leq r \leq$ 10 μm. Furthermore, pore networks with smaller pore-throat radii exhibit greater slopes in the $S_w - k_{rw}$ plots, as shown in Fig. 4.14(a), whereas S_{wc} remains scattered regardless of the pore-throat radii; see Fig. 4.14(b).

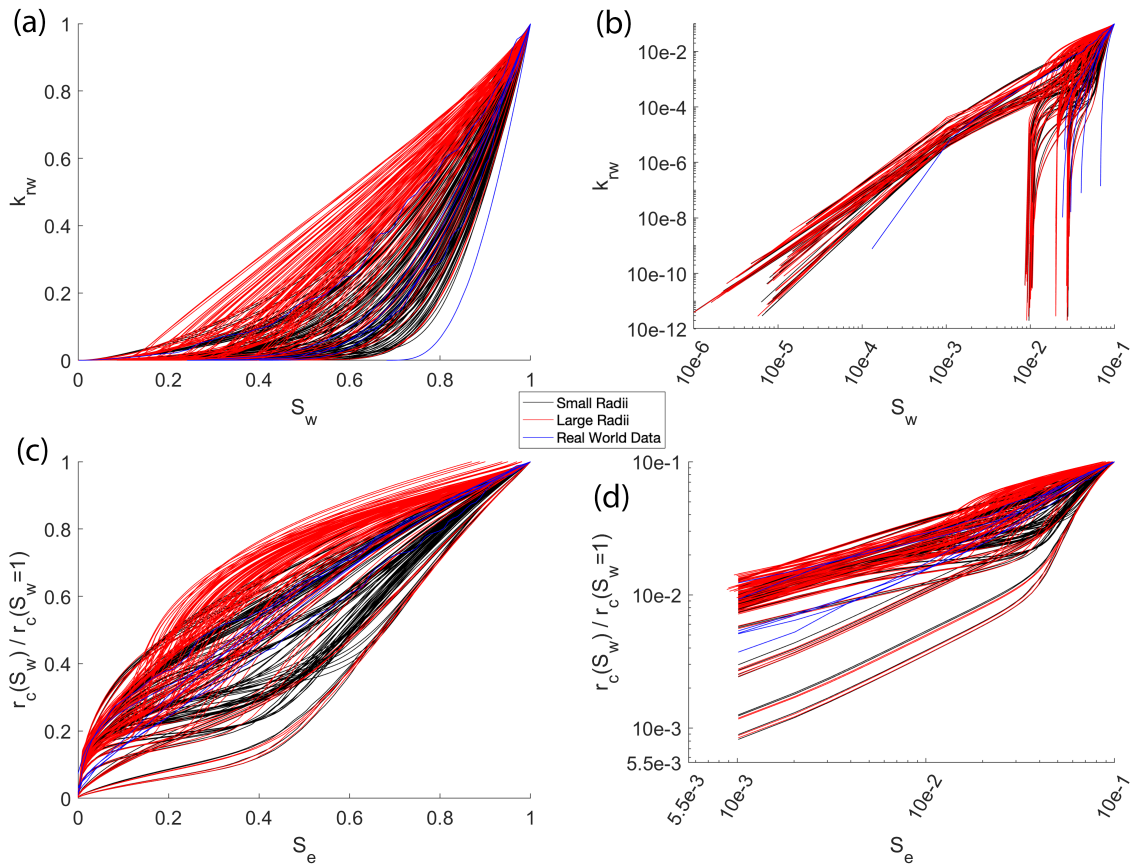

Figure 4.14: *Complete data set broken apart by pore throat radii.*

By further analysis of the pore-throat sizes we found an increasing trend between the permeability and critical pore-throat radius, as shown in Fig. 4.15(a), in accord with the results of Katz and Thompson (1986), Nishiyama and Yokoyama (2017), Ghanbarian et al.

(2017), and Ghanbarian et al. (2019). Figure 4.15(a) a plot of the permeability versus the critical pore-throat radius for three pore coordination numbers, $Z = 2$, 4, and 6. The data appear scattered, with higher permeabilities for networks with $Z = 6$ and lower values for $Z = 2$. The effect of pore coordination number on the permeability has been well addressed in the literature. For example, An et al. (2016) showed that pore networks with larger coordination numbers corresponded to higher permeabilities, which is of course expected. By incorporating the influence of formation factor, we show a definitive relationship between the critical pore size and the permeability. The trends shown in Fig. 4.15(b) confirm the importance of the critical pore-throat radius and the formation factor in the estimating permeability (An et al., 2016; Ghanbarian et al., 2019; Nishiyama and Yokoyama, 2017). Our results demonstrate that although the data in Fig. 4.15(a) are scattered, they do collapse onto each other in Fig. 4.15(b), where the permeability is plotted versus $r_c(S_w = 1)^2/F$. We did not find, however, any strong correlation between the formation factor and $r_c(S_w = 1)$, or between the porosity and $r_c(S_w = 1)$; see Figs. 4.15(c) and 4.15(d).

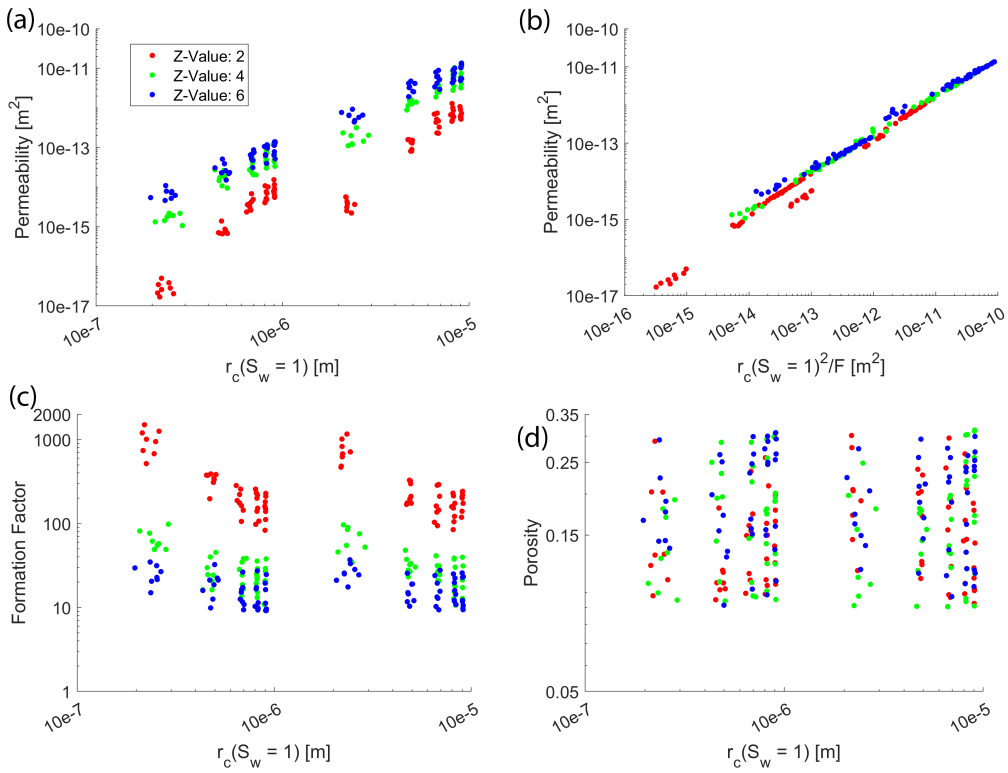

Figure 4.15: *Comparison of* $r_c(S_w = 1)$ *data with permeability* (a)*, formation factor* (c)*, and porosity,* (d)*. Plot* (b) *shows the relationship between the* $r_c(S_w = 1)^2/F$ *and permeability, following concepts from An et al. (2016), Nishiyama and Yokoyama (2017), and Ghanbarian et al. (2019).*

4.6 Study Limitations

Our proposed method for rock typing based on two-phase flow data and critical-path analysis (CPA) has some limitations. The CPA can only be used to model the wetting-phase relative permeability in porous rocks, because the wetting and non-wetting phases occupy, respectively, the smallest and largest pores. The smallest pores are normally ignored by the CPA, whereas it is the largest pores that are important to the CPA. As an alternative, one may apply the effective-medium approximation (Ghanbarian et al., 2016b; Levine and Cuthiell, 1986) to perform rock typing based on the wetting- and/or nonwetting-phase relative permeabilities. Nonetheless, the proposed approach in this study can group rocks in oil-wet

reservoirs using oil relative permeabilities, or in water-wet formations using water relative permeabilities. There exist evidence in the literature that the relative permeabilities are stress-dependent (Alexis et al., 2015; Huo and Benson, 2016; Lei et al., 2018; Ojagbohunmi et al., 2012). Depending on the mineralogical composition, rocks exhibit various elastic properties and, thus, distinct stress-dependent behaviors. Accordingly, the influence of the stress, which is not addressed by our proposed method, may be incorporated for rock-typing purposes, if the relative permeabilities are measured under in-situ confining pressure.

Chapter 5

Conclusion

Rock typing plays a critical role in reservoir engineering, particularly the accurate identification of rock types within a hydrocarbon reservoir. The accurate identification of rock types greatly improves reservoir characterization, ultimately increasing hydrocarbon production. This study aimed to develop and test a novel two-phase rock typing method centered around concepts form critical path analysis. To do this we simulated both single-phase and two-phase data using pore network modeling techniques, and created a large data set of petrophysical data that covered a comprehensive breadth of permeability, formation factor, porosity, and relative permeability curves. In addition to this synthetic database we also simulated six sandstone samples, one Mt. Simon sandstone, one Berea sandstone, and four Fontaineblaeu sandstones.

The use of our simulated data and our newly developed rock typing method resulted in the determination of twelve representative rock types. When comparing this to rock types found when using single-phase rock typing methods, we find that our two-phase method results in fewer rock types, twelve compared to the fifteen found when using Eq. 4.1, and more when compared to the eight clusters found using modified Archie's law (Eq. 2.1). Samples found within the same rock types for two-phase rock typing appear to be in different clusters when compared to single phase rock typing. These discrepancies among rock types clearly illustrate the importance of two-phase rock typing, especially when two-phase data is readily

available.

Future studies should be conducted using our rock typing method on experimental data sets to confirm the technical use of our rock typing method. Following this, further studies should be completed using our method in conjunction with reservoir modeling to test the accuracy of our method compared to other well known single-phase and two-phase methods.

Bibliography

Ahmed, T. (2001). *Reservoir Engineering Handbook, Second Edition*. Gulf Professional Publishing.

Alexis, D. A., Karpyn, Z. T., Ertekin, T., and Crandall, D. (2015). Fracture permeability and relative permeability of coal and their dependence on stress conditions. *Journal of Unconventional Oil and Gas Resources*, 10:1–10.

Aliakbardoust, E. and Rahimpour-Bonab, H. (2013). Integration of rock typing methods for carbonate reservoir characterization. *Journal of Geophysics and Engineering*, 10.

Alreshedan, F. and Kantzas, A. (2016). Investigation of permeability, formation factor, and porosity relationships for Mesaverde tight gas sandstones using random network models. *Journal of Petroleum Exploration and Production Technology*, 6(3):545–554.

Amaefule, J. O., Altunbay, M., Tiab, D., Kersey, D. G., and Keelan, D. K. (1993). Enhanced reservoir description: using core and log data to identify hydraulic (flow) units and predict permeability in uncored intervals/ wells. In *68th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers*, pages 205–220, Houston, TX.

An, S., Yao, J., Yang, Y., Zhang, L., Zhao, J., and Gao, Y. (2016). Influence of pore structure parameters on flow characteristics based on a digital rock and the pore network model. *Journal of Natural Gas Science and Engineering*, 31:156–163.

Anderson, W. G. (1987). Wettability Literature Survey - Part 5: the Effects of Wettability on Relative Permeability. *JPT, Journal of Petroleum Technology*, 39(11):1453–1468.

Archie, G. E. (1950). Introduction to petrophysics of reservoir rocks. *Bulletin of the American Association of Petroleum Geologists*, 34:943–961.

Arns, J. Y., Arns, C. H., Sheppard, A. P., Sok, R. M., Knackstedt, M. A., and Pinczewski, W. V. (2003). Relative permeability from tomographic images; effect of correlated heterogeneity. *Journal of Petroleum Science and Engineering*, 39(3-4):247–259.

Bashtani, F., Maini, B., and Kantzas, A. (2016). Single-phase and two-phase flow properties of mesaverde tight sandstone formation; random-network modeling approach. *Advances in Water Resources*, 94:174–184.

Blunt, M. J. (1997). Effects of heterogeneity and wetting on relative permeability using pore level modeling. *SPE Journal*, 2:70–87.

Bourbie, T. and Zinszner, B. (1985). Hydraulic and acoustic properties as a function of porosity in Fontainebleau Sandstone. *Journal of Geophysical Research*, 90(4):524–532.

Byrnes, A., Cluff, R., Webb, J., Victorine, J., Stalder, K., Osburn, D., Knoderer, A., Metheny, O., Hommertzheim, T., Byrnes, J., Krygowski, D., and Whittaker, S. (2008). Analysis of critical permeabilty, capillary pressure and electrical properties for mesaverde tight gas sandstones from western u.s. basins. *Journal of Petroleum Exploration and Production Technology*.

Compan, A. L., Bodstein, G. C., and Couto, P. (2016). A relative permeability rock-typing methodology with a clustering method combined with a heuristic optimization procedure. *SPE Journal*, 21:1899–1915.

Dashtian, H., Yang, Y., and Sahimi, M. (2015). Nonuniversality of the Archie exponent due to multifractality of resistivity well logs. *Geophysical Research Letters*, 42:10655–10662.

Dicarlo, D. A., Sahni, A., and Blunt, M. J. (2000). The effect of wettability on three-phase relative permeability. *Transport in Porous Media*, 39:347–366.

Faramarzi-Palangar, M. and Mirzaei-Paiaman, A. (2020a). Investigating dynamic rock quality in two-phase flow systems using tem-function: A comparative study of different rock typing indices. *Petroleum Research*, pages 1–10.

Faramarzi-Palangar, M. and Mirzaei-Paiaman, A. (2020b). Investigating dynamic rock quality in two-phase flow systems using TEM-function: A comparative study of different rock typing indices. *Petroleum Research*, pages 1–10.

Farshi, M., Moussavi-Harami, R., Mahboubi, A., Khanehbad, M., and Golafshani, T. (2019). Reservoir rock typing using integrating geological and petrophysical properties for the Asmari Formation in the Gachsaran oil field, Zagros basin. *Journal of Petroleum Science and Engineering*, 176(December 2018):161–171.

Gaffney, S. J. (2004). *Probabilistic curve-aligned clustering and prediction with regression mixture models*. PhD thesis, University of California - Irvine.

Ghanbarian, B. (2020). Applications of critical path analysis to uniform grain packings with narrow conductance distributions: II. Water relative permeability. *Advances in Water Resources*, 137(March 2019):103524.

Ghanbarian, B. and Hunt, A. G. (2017). Improving unsaturated hydraulic conductivity estimation in soils via percolation theory. *Geoderma*, 303:9–18.

Ghanbarian, B., Hunt, A. G., and Daigle, H. (2016a). Fluid flow in porous media with rough pore-solid interface. *Water Resources Research*, 52:2045–2058.

Ghanbarian, B., Hunt, A. G., Skaggs, T. H., and Jarvis, N. (2017). Upscaling soil saturated hydraulic conductivity from pore throat characteristics. *Advances in Water Resources*, 104:105–113.

Ghanbarian, B., Lake, L. W., and Sahimi, M. (2019). Insights into rock typing: A critical study. *SPE Journal*, 24(1):230–242.

Ghanbarian, B., Sahimi, M., and Daigle, H. (2016b). Modeling relative permeability of water in soil: Application of effective-medium approximation and percolation theory. *Water Resources Research*, 57:5025–5040.

Ghanbarian, B., Torres-Verdín, C., and Skaggs, T. H. (2016c). Quantifying tight-gas sandstone permeability via critical path analysis. *Advances in Water Resources*, 92:316–322.

Ghanbarian-Alavijeh, B. and Hunt, A. G. (2012). Unsaturated hydraulic conductivity in porous media: Percolation theory. *Geoderma*, 187:77–84.

Gregorio, J., Gonzalez, R., Mohammed, F., Blooshi, A., Ahmed, M., and Teneiji, A. (2020). Three rock-typing methods and implementation as part of the reservoir characterization and uncertainty assessment: An example from the arab formation upper jurassic, onshore field united arab emirates. In *Abu Dhabi International Petroleum Exhibition & Conference*, Abu Dhabi, UAE.

Gunter, G. W., Pinch, J. J., Finneran, J. M., and Bryant, W. T. (1997). Overview of an integrated process model to develop petrophysical based reservoir descriptions. *Society or Petroleum Engineers*, pages 475–479.

Hamon, G. and Bennes, M. (2004). Two-phase flow rock typing: Another approach. *Petrophysics*, 45:433–444.

Hao, L. and Cheng, P. (2010). Pore-scale simulations on relative permeabilities of porous media by lattice Boltzmann method. *International Journal of Heat and Mass Transfer*, 53:1908–1913.

Hollis, C., Vahrenkamp, V., Tull, S., Mookerjee, A., Taberner, C., and Huang, Y. (2010). Pore system characterisation in heterogeneous carbonates: An alternative approach to widely-used rock-typing methodologies. *Marine and Petroleum Geology*, 27(4):772–793.

Hunt, A. G. (2001). Applications of percolation theory to porous media with distributed local conductances. *Advances in Water Resources*, 24(3-4):279–307.

Huo, D. and Benson, S. M. (2016). Experimental investigation of stress-dependency of relative permeability in rock fractures. *Transport in Porous Media*, 113:567–590.

Johnson, D. L., Koplik, J., and Schwartz, L. M. (1986). New pore-size parameter characterizing transport in porous media. *Physical Review Letters*, 57:2564–2567.

Kadkhodaie, A. and Kadkhodaie, R. (2018). A Review of Reservoir Rock Typing Methods in Carbonate Reservoirs: Relation between Geological, Seismic, and Reservoir Rock Types. *Petroleum Engineering Iranian Journal of Oil & Gas Science and Technology*, 7(4):13–35.

Katz, A. J. and Thompson, A. H. (1986). Quantitative prediction of permeability and electrical conductivity in porous rock. In *1986 SEG Annual Meeting, SEG 1986*, pages 6–7. Society of Exploration Geophysicists.

Kohanpur, A. H., Rahromostaqim, M., Valocchi, A. J., and Sahimi, M. (2020). Two-phase flow of CO2-brine in a heterogeneous sandstone: Characterization of the rock and comparison of the lattice-Boltzmann, pore-network, and direct numerical simulation methods. *Advances in Water Resources*, 135(August 2019).

Kolodzie, S. (1980). Analysis of pore throat size and use of the Waxman-Smits Equation to determine OOIP in Spindle Field, Colorado. In *55th Annual Fall Technical Conference and Exhibition of the Society of Petroleum Engineers of AIME*, Dallas, Texas.

Landry, C. J., Karpyn, Z. T., and Ayala, O. (2014). Relative permeability of homogenous-wet and mixed-wet porous media as determined by pore-scale lattice Boltzmann modeling. *Water Resources Research*, 50:3672–3689.

Lei, G., Mo, S., Dong, Z., Wang, C. A. I., and Li, W. (2018). Theoretical and experimental study on stress-dependency of oil–water relative permeability in fractal porous media. *Fractals*, 26:1840010.

Levine, S. and Cuthiell, D. L. (1986). Relative permeabilities in two-phase flow through porous media: An application of effective medium theory. *J. Can. Pet. Technol.*, 25:74–84.

Li, H., Pan, C., and Miller, C. T. (2005). Pore-scale investigation of viscous coupling effects for two-phase flow in porous media. *Physical Review E*, 72:026705.

Lindquist, W. B., Venkatarangan, A., Dunsmuir, J. R., and Wong, T.-f. (2000). synchrotron X-ray tomographic images. *Journal of Geophysical Research*, 105:509–521.

Liu, Y., Liu, Y., Zhang, Q., Li, C., Feng, Y., Wang, Y., Xue, Y., and Ma, H. (2019). Petrophysical static rock typing for carbonate reservoirs based on mercury injection capillary pressure curves using principal component analysis. *Journal of Petroleum Science and Engineering*, 181(January):106175.

Lucia, F. (1995). Rock-fabric/petrophysical classification of carbonate pore space for reservoir characterization. *Bulletin of the American Association of Petroleum Geologists*.

Mahmud, W. M., Arns, J. Y., Sheppard, A., Knackstedt, M. A., and Pinczewski, W. V. (2007). Effect of network topology on two-phase imbibition relative permeability. *Transport in Porous Media*, 66(3):481–493.

Martin, A. J., Solomon, S. T., and Hartmann, D. J. (1996). Characterization of petrophysical flow units in carbonate reserviors. *The American Association of Petroleum Geologists*, 81(5):734–759.

Michel, R. and Bruno, L. (2014). Rock-typing In Carbonates: A Critical Review Of Clustering Methods. In *Abu Dhabi International Petroleum Exhibition and Conference*, Abu Dhabi, UAE. Society of Petroleum Engineers.

Mirzaei-Paiaman, A. and Ghanbarian, B. (2020). A new methodology for grouping and averaging capillary pressure curves for reservoir models. *Energy Geoscience*, 2(1):52–62.

Mirzaei-Paiaman, A. and Ghanbarian, B. (2021). A note on dynamic rock typing and TEM-function for grouping, averaging and assigning relative permeability data to reservoir simulation models. *Journal of Natural Gas Science and Engineering*, 87(December 2020):103789.

Mirzaei-Paiaman, A., Ostadhassan, M., Rezaee, R., Saboorian-Jooybari, H., and Chen, Z. (2018). A new approach in petrophysical rock typing. *Journal of Petroleum Science and Engineering*, 166:445–464.

Mirzaei-Paiaman, A., Sabbagh, F., Ostadhassan, M., Shafiei, A., Rezaee, R., Saboorian-Jooybari, H., and Chen, Z. (2019a). A further verification of fzi* and psrti: Newly developed petrophysical rock typing indices. *Journal of Petroleum Science and Engineering*, pages 693–705.

Mirzaei-Paiaman, A., Saboorian-Jooybari, H., Chen, Z., and Ostadhassan, M. (2019b). New technique of true effective mobility (tem-function) in dynamic rock typing: Reduction of uncertainties in relative permeability data for reservoir simulation. *Journal of Petroleum Science and Engineering*, 179:210–227.

Mirzaei-Paiaman, A., Saboorian-Jooybari, H., and Pourafshary, P. (2015). Improved Method to Identify Hydraulic Flow Units for Reservoir Characterization. *Energy Technology*, 3(7):726–733.

Müller-Huber, E., Schön, J., and Börner, F. (2015). The effect of a variable pore radius on formation resistivity factor. *Journal of Applied Geophysics*, 116:173–179.

Nelson, P. H. (1994). Permeability-porosity relationships in sedimentary rocks. *The Log Analyst*.

Nishiyama, N. and Yokoyama, T. (2017). Permeability of porous media: Role of the critical pore size. *Journal of Geophysical Research: Solid Earth*, 122(9):6955–6971.

Ojagbohunmi, S., Chalaturnyk, R., and Leung, J. (2012). Coupling of stress dependent relative permeability and reservoir simulation. In *SPE Symposium on Improved Oil Recovery*, pages 1–12, Tulsa, OK.

Pittman, E. D. (1992). Relationship of porosity and permeability to various parameters derived from mercury injection-capillary pressure curves for sandstone. *American Association of Petroleum Geologists Bulletin*, 76:191–198.

Porter, C. R. and Carothers, J. E. (1971). Porosity Relation Derived From Well Log Data. *The Log Analyst*, pages 16–26.

Rezaee, R., Saeedi, A., and Clennell, B. (2012). Tight gas sands permeability estimation from mercury injection capillary pressure and nuclear magnetic resonance data. *Journal of Petroleum Science and Engineering*, 88-89:92–99.

Riazi, Z. (2018). Application of integrated rock typing and flow units identification methods for an Iranian carbonate reservoir. *Journal of Petroleum Science and Engineering*, 160(May 2016):483–497.

Rushing, J., Newsham, K., and Blasingame, T. (2008). Rock Typing-Keys to Understanding Productivity in Tight Gas Sands. In *2008 SPE Unconventional Reservoirs Conference*, pages 1–31, Keystone, CO.

Sen, P. N., Goode, P. A., and Sibbit, A. (1988). Electrical conduction in clay bearing sandstones at low and high salinities. *Journal of Applied Physics*, 63(10):4832–4840.

Sen, P. N., Straley, C., Kenyon, W. E., and Whittingham, M. S. (1990). Surface-to-volume ratio, charge density, nuclear magnetic relaxation, and permeability in clay-bearing sandstones. *Geophysics*, 55(1):61–69.

Skalinski, M. and Kenter, J. A. (2015). Carbonate petrophysical rock typing: Integrating geological attributes and petrophysical properties while linking with dynamic behaviour. *Geological Society Special Publication*, 406:229–259.

Valvatne, P. H. (2004). *Predictive pore-scale modelling of multiphase flow.* PhD thesis, Imperial College of London.

Winsauer, W. O., Jr., H. M. S., Masson, P. H., and Williams, M. (1952). Resistivity of brine-saturated sands in relation to pore geometry. *AAPG Bulletin*, 36:253–277.

Xu, W., Ok, J. T., Xiao, F., Neeves, K. B., and Yin, X. (2014). Effect of pore geometry and interfacial tension on water-oil displacement efficiency in oil-wet microfluidic porous media analogs. *Physics of Fluids*, 26(9).

Appendix A

Total Simulation data set

Table A.1: Complete simulation data set

Sample	Porosity	Permeability $[m^2]$	Formation Factor	Critical Pore Radius $[m]$
1	0.123342	7.05E-14	181.0374	9.01E-06
2	0.095318	3.37E-13	37.82428	9.09E-06
3	0.264868	1.40E-12	9.210714	9.12E-06
4	0.106171	6.04E-14	210.9835	9.06E-06
5	0.298657	1.19E-12	10.88932	8.90E-06
6	0.135356	6.56E-13	19.49437	8.99E-06
7	0.297549	1.54E-13	83.012	8.98E-06
8	0.190306	6.16E-13	20.66398	9.01E-06
9	0.306829	1.39E-12	9.204659	9.05E-06
10	0.165158	8.42E-14	150.4737	9.04E-06
11	0.189915	6.13E-13	20.77847	8.99E-06
12	0.294333	1.33E-12	9.616462	9.12E-06
13	0.105196	4.86E-14	217.2136	8.22E-06
14	0.178072	5.66E-13	19.33555	8.10E-06

15	0.098533	4.00E-13	27.27537	8.21E-06
16	0.162755	7.65E-14	139.9118	8.17E-06
17	0.101152	3.03E-13	35.75339	8.19E-06
18	0.15559	6.60E-13	16.6237	8.10E-06
19	0.126691	5.20E-14	200.3062	8.17E-06
20	0.180059	4.93E-13	21.88717	8.19E-06
21	0.300053	1.17E-12	9.349922	8.26E-06
22	0.258018	1.08E-13	97.80766	8.06E-06
23	0.173829	4.72E-13	22.8551	8.19E-06
24	0.289285	1.12E-12	9.760612	8.29E-06
25	0.173861	4.90E-14	144.0316	6.96E-06
26	0.14914	3.26E-13	24.32219	6.88E-06
27	0.102548	3.10E-13	26.39629	6.92E-06
28	0.234501	6.82E-14	105.8879	6.82E-06
29	0.09781	2.06E-13	38.11747	6.98E-06
30	0.150922	4.76E-13	17.3061	6.92E-06
31	0.149391	3.61E-14	187.6252	6.48E-06
32	0.217706	4.19E-13	18.71217	6.90E-06
33	0.264717	7.56E-13	10.84844	7.03E-06
34	0.160238	3.85E-14	175.9125	6.67E-06
35	0.14633	2.74E-13	28.36691	6.66E-06
36	0.247204	7.00E-13	11.70457	6.89E-06
37	0.107089	8.02E-15	334.0912	4.96E-06
38	0.107987	1.08E-13	39.95886	4.74E-06
39	0.128485	2.20E-13	22.40233	5.14E-06
40	0.095176	7.08E-15	376.4615	4.54E-06
41	0.094313	9.49E-14	45.24545	5.05E-06

42	0.133726	2.33E-13	21.21069	5.20E-06
43	0.107253	6.91E-15	374.2144	4.59E-06
44	0.230994	2.08E-13	20.64002	4.85E-06
45	0.172228	2.61E-13	18.65441	4.95E-06
46	0.115404	7.56E-15	344.0174	4.94E-06
47	0.196024	1.76E-13	24.23483	4.90E-06
48	0.198951	3.06E-13	15.95274	4.34E-06
49	0.130236	2.59E-16	1008.256	2.25E-06
50	0.106804	1.32E-14	81.49202	2.08E-06
51	0.136631	6.20E-14	26.7355	2.66E-06
52	0.131043	2.83E-16	945.1072	2.50E-06
53	0.132488	1.92E-14	57.52447	2.60E-06
54	0.144183	5.22E-14	31.47424	2.53E-06
55	0.289675	5.00E-16	517.2705	2.24E-06
56	0.169831	1.94E-14	55.02982	2.55E-06
57	0.223812	7.91E-14	20.52023	2.38E-06
58	0.097867	1.68E-16	1501.632	2.20E-06
59	0.180224	2.20E-14	48.96776	2.46E-06
60	0.292323	1.09E-13	14.98391	2.36E-06
61	0.187307	1.08E-11	118.718	8.92E-05
62	0.258494	1.01E-10	12.82356	9.02E-05
63	0.114719	5.49E-11	23.24441	9.11E-05
64	0.09279	5.29E-12	240.5585	9.00E-05
65	0.114437	4.09E-11	31.23789	9.08E-05
66	0.117312	5.60E-11	22.78927	9.02E-05
67	0.141685	7.28E-12	173.9116	9.09E-05
68	0.313152	1.04E-10	12.25595	9.06E-05

69	0.268663	1.21E-10	10.58432	9.10E-05
70	0.178346	9.07E-12	139.9112	8.98E-05
71	0.310766	1.04E-10	12.3635	9.03E-05
72	0.235766	1.05E-10	12.17762	9.02E-05
73	0.265007	1.29E-11	84.48645	8.09E-05
74	0.303709	1.04E-10	10.66139	8.33E-05
75	0.132872	5.55E-11	19.72778	8.20E-05
76	0.10675	4.95E-12	213.6125	8.12E-05
77	0.118247	3.60E-11	30.15206	8.28E-05
78	0.118427	4.90E-11	22.33353	8.26E-05
79	0.166211	6.85E-12	152.8618	8.12E-05
80	0.219674	6.06E-11	17.86173	8.28E-05
81	0.261813	1.01E-10	10.83123	8.36E-05
82	0.242287	1.01E-11	104.961	8.16E-05
83	0.250938	7.01E-11	15.46237	8.19E-05
84	0.233698	8.95E-11	12.20974	8.19E-05
85	0.240084	6.98E-12	103.4215	6.50E-05
86	0.138976	3.01E-11	26.26932	6.74E-05
87	0.258232	8.91E-11	9.376825	6.84E-05
88	0.182414	5.19E-12	136.692	6.69E-05
89	0.119291	2.54E-11	31.02523	6.88E-05
90	0.11271	3.41E-11	23.99645	6.54E-05
91	0.192694	4.72E-12	144.7276	6.83E-05
92	0.132859	2.47E-11	31.45906	6.73E-05
93	0.225599	6.35E-11	12.87313	6.77E-05
94	0.098576	2.34E-12	285.5539	6.71E-05
95	0.193296	3.72E-11	21.03696	6.77E-05

96	0.217481	6.13E-11	13.33754	6.70E-05
97	0.171318	1.30E-12	210.2716	4.94E-05
98	0.116738	1.15E-11	37.4303	4.70E-05
99	0.146096	2.62E-11	18.9302	4.99E-05
100	0.121767	8.99E-13	298.3117	4.92E-05
101	0.180737	1.89E-11	23.21028	4.79E-05
102	0.188733	3.40E-11	14.49876	4.79E-05
103	0.203423	1.35E-12	196.057	4.88E-05
104	0.178088	1.60E-11	26.72571	4.89E-05
105	0.258454	4.20E-11	11.70818	4.73E-05
106	0.228477	1.53E-12	174.3044	4.96E-05
107	0.144155	1.27E-11	33.47948	4.93E-05
108	0.29392	4.77E-11	10.34202	4.87E-05
109	0.206271	4.32E-14	622.5491	2.21E-05
110	0.15592	2.00E-12	54.77781	2.33E-05
111	0.139227	5.81E-12	28.36744	2.56E-05
112	0.276938	5.54E-14	482.153	2.19E-05
113	0.106104	1.44E-12	75.39894	2.71E-05
114	0.117693	4.42E-12	36.9518	2.42E-05
115	0.204584	3.87E-14	675.2397	2.22E-05
116	0.24845	3.17E-12	34.36229	2.49E-05
117	0.224425	7.70E-12	21.07905	2.07E-05
118	0.301541	5.66E-14	468.5755	2.18E-05
119	0.111405	1.25E-12	84.93003	2.35E-05
120	0.204455	6.61E-12	24.45992	2.69E-05
121	0.198614	1.16E-13	111.3416	9.05E-06
122	0.122356	4.39E-13	29.07133	9.01E-06

123	0.159464	7.86E-13	16.28758	9.04E-06
124	0.138793	7.97E-14	160.5599	8.96E-06
125	0.131443	4.76E-13	26.86079	9.02E-06
126	0.108579	5.17E-13	24.67595	9.12E-06
127	0.109078	5.52E-14	228.6772	9.06E-06
128	0.201186	6.52E-13	19.53772	9.05E-06
129	0.307488	1.40E-12	9.163343	9.08E-06
130	0.178536	9.12E-14	139.182	9.04E-06
131	0.195283	6.32E-13	20.15558	9.04E-06
132	0.254338	1.14E-12	11.23355	9.03E-06
133	0.14988	7.01E-14	152.2738	8.24E-06
134	0.141115	4.36E-13	24.96984	8.14E-06
135	0.246127	1.12E-12	9.900264	8.27E-06
136	0.21167	1.01E-13	106.8954	8.36E-06
137	0.122674	3.75E-13	28.95991	8.22E-06
138	0.149606	6.34E-13	17.31507	8.12E-06
139	0.099047	4.04E-14	256.8643	8.04E-06
140	0.257482	7.17E-13	15.12994	8.23E-06
141	0.249809	9.60E-13	11.38596	8.29E-06
142	0.109513	4.49E-14	231.5579	8.12E-06
143	0.293273	8.29E-13	13.12804	8.19E-06
144	0.274636	1.06E-12	10.31424	8.00E-06
145	0.154434	4.30E-14	162.8757	6.82E-06
146	0.097111	2.04E-13	38.43438	7.21E-06
147	0.156811	5.03E-13	16.41072	6.84E-06
148	0.114395	3.12E-14	221.4163	6.67E-06
149	0.107492	2.26E-13	34.73114	6.87E-06

150	0.16771	5.44E-13	15.18394	6.81E-06
151	0.109232	2.61E-14	256.7142	6.76E-06
152	0.287542	5.75E-13	13.75206	6.85E-06
153	0.231716	6.52E-13	12.53933	6.73E-06
154	0.099559	2.36E-14	283.37	6.42E-06
155	0.243358	4.72E-13	16.64488	6.98E-06
156	0.299198	8.76E-13	9.386758	6.99E-06
157	0.181212	1.39E-14	197.1762	4.71E-06
158	0.143705	1.46E-13	29.7814	4.57E-06
159	0.091576	1.51E-13	32.41691	4.99E-06
160	0.117083	8.67E-15	309.2299	4.91E-06
161	0.249802	2.74E-13	16.22028	4.34E-06
162	0.263542	5.11E-13	9.874056	4.76E-06
163	0.102761	6.79E-15	382.9961	5.06E-06
164	0.194194	1.74E-13	24.59539	4.60E-06
165	0.250624	3.90E-13	12.5851	4.88E-06
166	0.10207	6.64E-15	389.4287	4.76E-06
167	0.287357	2.66E-13	16.25081	4.84E-06
168	0.154396	2.30E-13	21.19134	4.71E-06
169	0.109517	2.04E-16	1257.377	2.60E-06
170	0.127305	1.78E-14	61.56473	2.41E-06
171	0.176308	7.28E-14	22.80505	2.54E-06
172	0.202657	3.85E-16	678.3971	2.46E-06
173	0.099865	1.42E-14	76.6537	2.33E-06
174	0.189961	7.73E-14	21.5227	2.55E-06
175	0.202977	3.46E-16	739.7563	2.16E-06
176	0.094968	1.07E-14	98.19577	2.90E-06

177	0.143567	4.62E-14	34.79588	2.35E-06
178	0.121102	2.12E-16	1199.211	2.14E-06
179	0.192061	2.18E-14	49.08643	2.82E-06
180	0.166213	5.45E-14	29.57587	1.96E-06
181	0.101505	5.79E-12	219.7178	9.03E-05
182	0.091109	3.21E-11	39.68223	9.01E-05
183	0.10438	4.95E-11	25.79841	8.99E-05
184	0.110824	6.34E-12	200.9671	8.99E-05
185	0.164018	6.04E-11	21.21626	9.01E-05
186	0.253188	1.32E-10	9.721133	9.06E-05
187	0.119056	6.07E-12	208.1165	8.98E-05
188	0.226251	7.40E-11	17.24424	9.04E-05
189	0.242824	1.08E-10	11.79714	9.02E-05
190	0.102526	5.22E-12	241.8064	9.04E-05
191	0.171906	5.53E-11	23.02091	9.06E-05
192	0.299538	1.36E-10	9.414715	9.09E-05
193	0.143854	6.72E-12	158.5953	7.96E-05
194	0.093942	2.79E-11	38.77752	8.19E-05
195	0.161625	6.91E-11	15.90699	8.24E-05
196	0.099207	4.60E-12	229.4318	8.07E-05
197	0.096151	2.87E-11	37.64052	8.19E-05
198	0.106521	4.35E-11	25.08052	8.12E-05
199	0.208387	8.66E-12	121.6191	8.22E-05
200	0.30255	8.57E-11	12.69796	8.16E-05
201	0.240321	9.22E-11	11.85736	8.21E-05
202	0.145454	6.00E-12	174.0187	8.25E-05
203	0.212504	5.86E-11	18.4453	8.23E-05

204	0.201753	7.60E-11	14.35358	8.33E-05
205	0.155388	4.34E-12	161.581	6.64E-05
206	0.090662	1.90E-11	41.25479	6.63E-05
207	0.135156	4.22E-11	19.46395	6.72E-05
208	0.120237	3.28E-12	210.9517	6.90E-05
209	0.131657	2.83E-11	27.87983	6.86E-05
210	0.097311	2.95E-11	27.74199	6.95E-05
211	0.096793	2.29E-12	291.281	6.84E-05
212	0.197994	3.79E-11	20.63253	6.85E-05
213	0.188834	5.23E-11	15.60193	6.93E-05
214	0.296168	7.40E-12	94.13373	6.73E-05
215	0.167314	3.17E-11	24.61126	6.92E-05
216	0.275828	7.99E-11	10.26955	6.62E-05
217	0.20096	1.60E-12	175.8384	4.73E-05
218	0.09089	8.91E-12	48.02914	4.63E-05
219	0.214489	4.17E-11	12.06083	5.11E-05
220	0.110504	8.13E-13	327.9554	4.86E-05
221	0.131113	1.34E-11	32.31255	4.87E-05
222	0.114629	1.93E-11	25.56706	4.74E-05
223	0.237875	1.58E-12	168.6487	4.69E-05
224	0.17461	1.53E-11	27.77278	4.78E-05
225	0.211079	3.26E-11	15.02697	4.77E-05
226	0.124425	8.09E-13	321.5227	4.91E-05
227	0.155988	1.40E-11	30.35515	5.20E-05
228	0.169207	2.56E-11	19.03065	5.03E-05
229	0.116826	2.22E-14	1162.707	2.34E-05
230	0.091362	1.11E-12	96.49444	2.24E-05

231	0.173426	6.58E-12	25.13821	2.27E-05
232	0.19063	3.66E-14	713.1374	2.43E-05
233	0.097611	1.20E-12	89.62547	2.37E-05
234	0.162102	6.43E-12	25.67823	2.27E-05
235	0.145215	2.51E-14	1013.95	2.21E-05
236	0.19797	2.35E-12	45.78475	2.11E-05
237	0.149383	4.83E-12	33.30207	2.43E-05
238	0.172341	3.12E-14	826.5165	2.21E-05
239	0.180206	2.05E-12	52.28994	2.89E-05
240	0.274737	9.28E-12	17.56293	2.37E-05
241	0.074669	1.68E-13	174.0523	1.65E-05
242	0.127726	7.40E-13	69.46686	2.09E-05
243	0.144163	1.29E-12	51.39044	2.39E-05
244	0.223781	2.00E-12	30.05941	1.83E-05
245	0.183296	2.52E-12	14.9653	1.74E-06
246	0.255502	4.15E-12	5.70859	6.35E-06

Appendix B

Code Database

Pore size distribution fitting

```
% Fits pore throat length distributions to distributions from lit
% Script written by Brandon Yokeley
% Script created on 27 May 2020
% Script was last updated 27 May 2020

% Fits pore throat length distributions created by a truncated weibull
% distribution as described in Valvatne, 2004 to pore throat length
% distrubtions gathered in the literature. Additionally, makes example
% distribution plots for thesis, to describe differences in pore size
% distributions.

```

```
12 %Closes all plots, clears all variables, and starts timer
13 close all
14 clear
15 tic
16
17
18 %% Data fitting
19
20 %Creates a vector of random values between 0 and 1 to create the
      weibull
21 %distribution
22 X = rand(100000,1);
23 %Sets minimum pore throat length
24 minPore = 0.1;
25 %Sets max pore throat length
26 maxPore = 10;
27
28 % dataDiff = nan(1883891,1);
29 % count = 0;
30 %
31 % %Loops through delta and Eta values until a match is found
32 % for deltaVal = 0.1:0.01:10
33 %     for etaVal = 1:0.01:20
34 %         count = count+1;
35 %
36 %         %Creates truncated weibull distribution
37 %         weibullDist = (maxPore - minPore) * (-deltaVal * log(X
```

```
.* (1-exp(-1/deltaVal)) + exp(-1/deltaVal))).^(1/etaVal) +
minPore;
%           %Normalizes frequency distribution
%           [N, E] = histcounts(weibullDist,length(litData));
%           N = N/length(X);
%
%           %Difference in data
%           dataDiff(count) =  nan();
%       end
% end

%Sets Delta value for distribution shape
deltaVal = 0.2;
%Sets eta value for distribution shape
etaVal = [24,12,6,3,1.35];

%Loops through each eta value
for i = 1:5
%Creates weibull distribution
weibullDist_24 = (maxPore - minPore) * (-deltaVal * log(X .* (1-
exp(-1/deltaVal)) + exp(-1/deltaVal))).^(1/etaVal(i)) + minPore
;
%Created weibull distribution
fig1 = figure(1);
subplot(1,2,1)
histogram(weibullDist_24, 100, 'Normalization', 'count')
hold on
```

```
%set(gca, 'XScale', 'log')
set(gca, 'FontSize', 14)
xlim([0, 10])
legend({'Eta: 24', 'Eta: 12', 'Eta: 6', 'Eta: 3', 'Eta: 1.35'}, 'Location', 'best')
xlabel('Pore Throart Radii [um]', 'FontSize', 18)
%sgtitle('Cummulative pore sizes', 'FontSize',24)
end

%Sets minimum pore throat length
minPore = 1;
%Sets max pore throat length
maxPore = 100;

%Loops through each eta value
for i = 1:5
%Creates weibull distribution
weibullDist_24 = (maxPore - minPore) * (-deltaVal * log(X .* (1-exp(-1/deltaVal)) + exp(-1/deltaVal))).^(1/etaVal(i)) + minPore;
%Created weibull distribution
fig1 = figure(1);
subplot(1,2,2)
histogram(weibullDist_24, 100, 'Normalization', 'count')
hold on
%set(gca, 'XScale', 'log')
set(gca, 'FontSize', 14)
```

```
xlim([1, 100])
legend({'Eta: 24', 'Eta: 12', 'Eta: 6', 'Eta: 3', 'Eta: 1.35'}, '
    Location', 'best')
xlabel('Pore Throart Radii [um]', 'FontSize', 18)
end


%Ends timmer
toc
```

Initial data set creation

```
% Creates pore networks to find permeability values for all samples
% Script created on 15 Novemeber 2019
% Script created by Brandon Yokeley
% Script updated on 5.15.2020
%
% Creates and runs netgen and poreflow code over all 240 samples. This
% code runs 10 iteration of each sample at a lattice size of 10, 20, 30,
% 40, 50, 60, and 65.

clear
close all
tic

%% Model Setup

%Lattice Size to calculate
latInc = [10,20,30,35,40,45,50,55,60,65];
%Marker colors for plots
markerColor = ['#E6194B'; '#f58231'; '#ffe119'; '#bfef45'; '#3cb44b'; '#42d4f4'; '#4363d8'; '#911eb4'; '#f032e6'; '#a9a9a9'];
%Number of iterations for for loop
itNum = 10;
```

```
%---Netgen input parameters
ng3 = load('throatRadi.csv');
ng4 = load('throatLength.csv');
ng5 = load('aspectRatio.csv');
ng6 = load('shapeFactor.csv');
ng7 = load('poreProportions.csv');
ng8 = load('throatProportions.csv');
ng9 = load('clayContent.csv');
ng10 = load('connNumber.csv');
ng11 = 'T';

%---Poreflow input parameters
pf1 = 'INIT_CON_ANG';
pf2 = load('contactAngle.csv');
pf3 = '#';
pf4 = 'NETWORK';
pf6 = '#';
pf7 = 'TITLE';
pf9 = '#';

%---Folder Structure
folderLoc = cd;
dataLoc = strcat(folderLoc(1:end-7), 'Data');
netgenLoc = [strcat(folderLoc,'\netgen_win32.exe'), ' ', strcat(
    folderLoc,'\netgenInput.dat')];
poreflowLoc = [strcat(folderLoc,'\poreflow_win32.exe'), ' ',
```

```
    strcat(folderLoc,'\poreflowInput.dat')];

%% Initial Parameters

%Test number
count = 0;
%Output Counter
outputCount = 0;
%Output data from poreflow simulation
outputData = nan(length(latInc)*itNum,6);
%Drain data, Sw, Kro, Krw
drainData = [];
%Single phase data
avgOutputData = nan(length(latInc),4);
%Distribution Data
poreRadi = [];
poreShape = [];
throatData =[];

%Relative permeability plots marker color
markerColor = ['#E6194B'; '#f58231'; '#ffe119'; '#bfef45'; '#3
    cb44b'];



%% Data Creation

```

```
%Start of the data creation loop that will loop through each model

%Pore size broadness
for i = 1:length(ng3)
    %Test number
    count = count+1;
    %Sets titles for netgen and poreflow input files
    ng1 = strcat('test',num2str(count));
    pf5 = strcat('test',num2str(count));
    pf8 = strcat('test',num2str(count));



    %Creates the working directory for this test
    testFolder = strcat(dataLoc,'\',ng1);
    mkdir(testFolder);
    mkdir(strcat(testFolder,'\Results'));

    %--Directort Creation - For Loop
    for n = 1:length(latInc)
        latFolder = strcat(testFolder,'\Lattice_',num2str(latInc(n
            )));
        mkdir(latFolder);
        for o = 1:itNum
            intFolder = strcat(latFolder, '\', num2str(o));
            mkdir(intFolder);
        end
```

```
    end

    %--Data Creation - For Loop
    for n = 1:length(latInc)

        %--Data Creation - Input Files
        %Writes netgen input file
        netgenWrite(ng1,latInc(n),ng3(i,:),ng4(i,:),ng5(i,:),ng6(i
            ,:),ng7(i,:),ng8(i,:),ng9(i),ng10(i),ng11);
        %Writes poreflow input file
        copyfile basePoreFlowInput.dat poreflowInput.dat
        poreflowWrite(pf1,pf2(i,:),pf3,pf4,pf5,pf6,pf7,pf8,pf9);
        %Lattice size
        latSize = latInc(n);
        %--Data Creation - Parallel loop
        for o = 1:itNum
            intFolder = strcat(dataLoc,'\',ng1,'\Lattice_',num2str
                (latSize),'\',num2str(o));
            cd(intFolder);
            system(netgenLoc);
            %!E:\rockTyping_Data\interporeScripts\netgen_win32.exe
                E:\rockTyping_Data\interporeScripts\netgenInput.
                dat
            cd(folderLoc);
        end

        parfor o = 1:itNum
```

```
            intFolder = strcat(dataLoc,'\',ng1,'\Lattice_',num2str(latSize),'\',num2str(o));
            cd(intFolder);
            system(poreflowLoc);
            %!E:\rockTyping_Data\interporeScripts\poreflow_win32.exe E:\rockTyping_Data\interporeScripts\poreflowInput.dat
            cd(folderLoc);
        end

        %--Printing Code
        %--Imports output data
        for o = 1:itNum
            %Sets iteration number
            outputCount = outputCount + 1;
            %States iteration folder number
            intFolder = strcat(dataLoc,'\',ng1,'\Lattice_',num2str(latSize),'\',num2str(o));
            %Creates temp variabels to import drain, output, and pore throat radii data
            %Drain data
            drainTemp = drainDataImport(strcat(intFolder,'\',ng1,'_draincycle_1.csv'));
            %Single Phase data
            outputTemp = outputDataImport(strcat(intFolder,'\',ng1,'.prt'));
            %Pore throat radi
```

```
            throatRadiTemp = throatRadiImport(strcat(intFolder,'\'
                ,ng1,'_link1.dat'));
            %Pore body radi and pore shape
            [poreRadiTemp, poreShapeTemp] = poreDataImport(strcat(
                intFolder,'\',ng1,'_node2.dat'));
            %Throat Length
            throatLengthTemp = throatLengthImport(strcat(intFolder
                ,'\',ng1,'_link2.dat'));
            %Adds lattice size and iteration number to drain data
            drainTemp = [ones(length(drainTemp), 1) * latSize,
                ones(length(drainTemp), 1) * o, drainTemp];

            %Saves pore throat length, radi, pore body radi, and
                pore body
            %shape distributions so they can be plotted
            throatData = [throatData; throatLengthTemp,
                throatRadiTemp];
            poreRadi = [poreRadi;poreRadiTemp];
            poreShape = [poreShape;poreShapeTemp];

            %Finds mode of pore throat radi distribution
            throatRadiTemp = mode(throatRadiTemp);
            %Adds imported drain and output data to variables to
                then be
            %written to new files
            drainData = [drainData; drainTemp];
            outputData(outputCount,:) = [count, latSize,
```

```
            outputTemp, throatRadiTemp];
        end


        %Averages single phase data so it can be plotted to find
            single phase
        %REV
        tempAVG = outputData(outputData(:,2) == latInc(n),:);
        avgOutputData(n,:) = [latInc(n), mean(tempAVG(:,3:5))];
        drainPlot = drainData(drainData(:,1) == latInc(n),:);

        %Color for plotting
        color = markerColor(n,:);

        %Relative Permeability Plots
        %Kro vs Sw
        fig1 = figure(1);
        hold on
        plot(drainPlot(:,3), drainPlot(:,6), 'o', 'MarkerFaceColor
            ', color, 'MarkerEdgeColor', color, 'MarkerSize', 3)
        xlabel('Water Saturation [-]', 'FontSize',18)
        ylabel('Relative Permability - Oil [mD]', 'FontSize',18)
        legend({'30','40','50','60','65'}, 'Location', 'best')
        title({'Test Number: ', count}, 'FontSize', 18)

        %Krw vs Sw
        fig2 = figure(2);
```

```
        hold on
        plot(drainPlot(:,3), drainPlot(:,5), 'o', 'MarkerFaceColor
            ', color, 'MarkerEdgeColor', color', 'MarkerSize', 3)
        xlabel('Water Saturation [-]', 'FontSize',18)
        ylabel('Relative Permability - Water [mD]', 'FontSize',18)
        legend({'30','40','50','60','65'}, 'Location', 'best')
        title({'Test Number: ', count}, 'FontSize', 18)

    end

    %---Plotting Code

    %Single phase REV plots
    fig3 = figure(3);
    sgtitle({'Test Number: ', count}, 'FontSize', 18)
    %sgtitle({'Pore Throat Length Parameters', num2str(ng4(i,:))},
        'FontSize',18)
    subplot(3,1,1)
    plot(avgOutputData(:,1), avgOutputData(:,2), '-or', '
        MarkerFaceColor','b', 'MarkerEdgeColor', 'b', 'LineWidth'
        ,1.25)
    ylabel('Porosity [-]', 'FontSize',16)
    ylim([0 1]);
    subplot(3,1,2)
    plot(avgOutputData(:,1), avgOutputData(:,3), '-or', '
        MarkerFaceColor','b', 'MarkerEdgeColor', 'b', 'LineWidth'
        ,1.25)
```

```
    ylabel('Permability [mD]', 'FontSize',16)
    subplot(3,1,3)
    plot(avgOutputData(:,1), avgOutputData(:,4), '-or', '
        MarkerFaceColor','b', 'MarkerEdgeColor', 'b', 'LineWidth'
        ,1.25)
    ylabel('Formation Factor [-]', 'FontSize',16)
    xlabel('Lattice Size', 'FontSize',16)
    set(gcf, 'Position', get(0, 'Screensize'));

    %Distribution plots
    fig4 = figure(4);
    sgtitle({'Test Number: ', count}, 'FontSize', 18)
    subplot(2,2,1);
    histogram(poreRadi)
    xlabel('Pore Body Radi [m]', 'FontSize',16)
    subplot(2,2,2);
    histogram(throatData(:,1))
    xlabel('Pore Throat Length [m]', 'FontSize',16)
    subplot(2,2,3);
    histogram(poreShape)
    xlabel('Pore Shape Factor [-]', 'FontSize',16)
    %Sets figures to be full screen for printing
    set(gcf, 'Position', get(0, 'Screensize'));
    subplot(2,2,4);
    histogram(throatData(:,2))
    xlabel('Pore Throat Radi [m]', 'FontSize',16)

```

```
%---Writes output data to output file

%Opens drain output file
fidDrain = fopen(strcat(ng1,'_Drain.txt'), 'w');
%Writes header for drain data file
fprintf(fidDrain, 'Test Number: %d\n', count);
fprintf(fidDrain, 'Lat Size   Iteration    Sw       PC
   Krw       Kro\n');
%Writes Drain Data file
for q = 1:length(drainData)
    fprintf(fidDrain, '%d %d %e %e %e %e\n', drainData(q,1),
       drainData(q,2), drainData(q,3), drainData(q,4),
       drainData(q,5),drainData(q,6));
end

%Opens results output file
fidOutput = fopen(strcat(ng1,'_Output.txt'), 'w');
%Writes header for output file
fprintf(fidOutput, 'Test Number: %d\n', count);
fprintf(fidOutput, 'Lat Size   Iteration   Porosity
   Permability   Form Factor\n');
%Writes output data into complete output file
for q = 1:outputCount
    fprintf(fidOutput, '%d %d %f %f %f %e\n',outputData(q,1),
       outputData(q,2),outputData(q,3),outputData(q,4),
       outputData(q,5),outputData(q,6));
end
```

```
    fclose('all');

    %--Output saving

    %Moves output data files to results folder
    movefile('poreflowInput.dat', strcat(dataLoc,'\',ng1,'\Results'));
    movefile('netgenInput.dat', strcat(dataLoc,'\',ng1,'\Results'));
    movefile(strcat(ng1,'_Output.txt'), strcat(dataLoc,'\',ng1,'\Results'));
    movefile(strcat(ng1,'_Drain.txt'), strcat(dataLoc,'\',ng1,'\Results'));

    %Saves plots and moves them to results folder
    print(fig1, strcat(ng1,'_kroPlot'), '-dpng', '-r300');
    print(fig2, strcat(ng1,'_krwPlot'), '-dpng', '-r300');
    print(fig3, strcat(ng1,'_outputPlot'), '-dpng', '-r300')
    print(fig4, strcat(ng1,'_distPlots'), '-dpng', '-r300');
    copyfile(strcat(ng1,'_kroPlot.png'), strcat(dataLoc,'\',ng1,'\Results'));
    copyfile(strcat(ng1,'_krwPlot.png'), strcat(dataLoc,'\',ng1,'\Results'));
    copyfile(strcat(ng1,'_outputPlot.png'), strcat(dataLoc,'\',ng1,'\Results'));
    copyfile(strcat(ng1,'_distPlots.png'), strcat(dataLoc,'\',ng1,'\Results'));
```

```
    %Deletes raw data to save disk space
    rmdir(strcat(dataLoc,'\',ng1,'\Lattice_*'), 's');

    %--Clears variables for next sample
    %Clears figures
    clf(fig1)
    clf(fig2)
    clf(fig3)
    clf(fig4)

    %Clears output variables for rewrite
    drainData = [];
    outputData = [];
    outputCount = 0;
    poreRadi = [];
    poreShape = [];
    throatData = [];


end

toc
```

Drain data import

```
function [drainData] = drainDataImport(filename)

%filename = 'poreTest6_04811_2_draincycle_1.csv';
fid = fopen(filename);

%Checks to make sure the filename is correct, if not it does not add any
%data to the inputed variable, and then ends the function
if fid == -1
    return
end

%Imports drain data from .csv file
temp = cell2mat(textscan(fid, '%f%f%f%f%f', 'Delimiter', ',', 'HeaderLines', 2));
%Adds imported drain data into the inputed variabe
drainData = [temp(:,1), temp(:,2), temp(:,3), temp(:,4)];

%Closes all open files
fclose('all');

end
```

Output data import

```
function [outputData] = outputDataImport(filename)%, outputData)

```

```
%Opens output file
fid = fopen(filename);

%Checks to make sure filename is correct, if it is not then it terminates
%the function
if fid == -1%
    outputData = outputData;
    return
end

%Imports porosity [-], absolute permeability [mD], and formation factor [-]
temp = textscan(fid, '%s', 18, 'HeaderLines', 58);
%Saves porosity, permeability, and formation factor to inputed variable
%outputData = [outputData; str2double(temp{1}(3)), str2double(temp{1}(11)), str2double(temp{1}(18))];
outputData = [str2double(temp{1}(3)), str2double(temp{1}(11)), str2double(temp{1}(18))];
%Closes all open files
fclose('all');

end
```

Throat radii import

```
function throatVar1 = throatRadiImport(filename)
```

```
%Opens File
fid = fopen(filename);

%Checks to make sure the filename is correct, if not it does not add any
%data to the inputed variable, and then ends the function
if fid == -1
    return
end

%Imports pore radi and pore shape factor from node file
poreData = cell2mat(textscan(fid, '%f %f %f %f %f %f %f %f', 'HeaderLines',1));
%Throat Length
throatVar1 = poreData(:,4);


%Closes the file
fclose('all');

end
```

Throat length import

```
function throatVar1 = throatLengthImport(filename)

%Opens File
```

```
fid = fopen(filename);

%Checks to make sure the filename is correct, if not it does not add any
%data to the inputed variable, and then ends the function
if fid == -1
    return
end

%Imports pore radi and pore shape factor from node file
poreData = cell2mat(textscan(fid, '%f %f %f %f %f %f %f %f'));
%Throat Length
throatVar1 = poreData(:,6);


%Closes the file
fclose('all');

end
```

Final data set creation

```
% Creates pore networks to find permeability values for all samples
% Script created on 25 August, 2020
% Script created by Brandon Yokeley
% Script updated on 8.25.2020
%
```

```
% Creates and runs netgen and poreflow code over all 240 samples. This
% code runs 100 iteration of each sample at a lattic size of 65


clear
close all
tic

%% Model Setup

%Lattice size to calculate at
latInc = 65;

%---Netgen input parameters
ng3 = load('throatRadi.csv');
ng4 = load('throatLength.csv');
ng5 = load('aspectRatio.csv');
ng6 = load('shapeFactor.csv');
ng7 = load('poreProportions.csv');
ng8 = load('throatProportions.csv');
ng9 = load('clayContent.csv');
ng10 = load('connNumber.csv');
ng11 = 'T';

%---Poreflow input parameters
%---Poreflow input parameters
```

```
pf1 = 'INIT_CON_ANG';
pf2 = load('contactAngle.csv');
pf3 = '#';
pf4 = 'NETWORK';
pf6 = '#';
pf7 = 'TITLE';
pf9 = '#';

%---Folder Structure
folderLoc = cd;
dataLoc = strcat(folderLoc(1:end-7), 'Data');

%% Initial Parameters

%Output counter
outputCount = 0;
%Output data from poreflow simulation
outputData = nan(100,6);
%Drain Data matrix
drainData = [];

%% Data Creation Loop

%Start of data creation
for i = 1:240
    %% Data Creation
    %Test number
```

```
    count = i;
    %Creates title for directory
    ng1 = strcat('test',num2str(count));

    %Creates the working directory for this test
    testFolder = strcat(dataLoc,'\',ng1);
    mkdir(testFolder);
    mkdir(strcat(testFolder,'\Results'));

    copyfile('basePoreFlowInput.dat', testFolder);
    copyfile('*win32.exe', testFolder)
    copyfile('dataLoop_*', testFolder);
    cd(testFolder);

    for j = 1:100
        %Sets test number title
        netgenName = strcat('netgenInput','_',num2str(j));
        ng1 = strcat('test',num2str(count),'_',num2str(j));
        pf5 = strcat('test',num2str(count),'_',num2str(j));
        pf8 = strcat('test',num2str(count),'_',num2str(j));
        %writes needed network files
        netgenWrite(netgenName,ng1,latInc,ng3(i,:),ng4(i,:),ng5(i
            ,:),ng6(i,:),ng7(i,:),ng8(i,:),ng9(i),ng10(i),ng11);
        %Writes needed poreflow input file
        copyfile('basePoreFlowInput.dat',strcat('poreflowInput','_
            ',num2str(j),'.dat'));
        poreflowWrite(num2str(j),pf1,pf2(i,:),pf3,pf4,pf5,pf6,pf7,
```

```
        pf8,pf9);
    end

    %Runs netgen and poreflow code in 25 iteration chunks until all 100
    %iterations are completed. Uses custom written batch files to run code
    system('dataLoop_1.bat 1>NUL 2>NUL');
    system('dataLoop_2.bat 1>NUL 2>NUL');
    system('dataLoop_3.bat 1>NUL 2>NUL');
    system('dataLoop_4.bat 1>NUL 2>NUL');
    system('dataLoop_5.bat 1>NUL 2>NUL');
    system('dataLoop_6.bat 1>NUL 2>NUL');
    system('dataLoop_7.bat 1>NUL 2>NUL');

    %% Printing Code
    %Imports all drain and output data for it to be written
    for j = 1:100
        %Sets iteration number for writing
        outputCount = outputCount + 1;
        %Sets drain data name
        ng1 = strcat('test',num2str(count),'_',num2str(j));
        %Creates a temp variable to store drain data in
        drainTemp = drainDataImport(strcat(ng1,'_draincycle_1.csv'));
        %Creates temp variable to store output data in
        outputTemp = outputDataImport(strcat(ng1,'.prt'));
```

```
        %Imports throat radi data to find mode of pore throat radi
        throatRadiTemp = throatRadiImport(strcat(ng1,'_link1.dat'));
        %Finds mode of pore throat radius
        throatRadiTemp = mode(throatRadiTemp);

        %Creates drain data matrix to be written to drain output file
        drainData = [drainData; ones(length(drainTemp), 1) * latInc, ones(length(drainTemp), 1) * j, drainTemp];
        %Creates output data matrix to be written too
        outputData(outputCount,:) = [count, latInc, outputTemp, throatRadiTemp];
    end

    %Sets title for results files
    ng1 = strcat('test',num2str(count));

    %-----Writes Drain Data to Text File
    %Opens drain output file
    fidDrain = fopen(strcat(ng1,'_Drain.txt'), 'w');
    %Writes header for drain data file
    fprintf(fidDrain, 'Test Number: %d\n', count);
    fprintf(fidDrain, 'Lat Size    Iteration    Sw        PC        Krw       Kro\n');
    %Writes Drain Data file
    for q = 1:length(drainData)
```

```
        fprintf(fidDrain, '%d %d %e %e %e %e\n', drainData(q,1),
            drainData(q,2), drainData(q,3), drainData(q,4),
            drainData(q,5),drainData(q,6));
    end

    %------Writes Output Data to Text File
    %Opens results output file
    fidOutput = fopen(strcat(ng1,'_Output.txt'), 'w');
    %Writes header for output file
    fprintf(fidOutput, 'Test Number: %d\n', count);
    fprintf(fidOutput, 'Lat Size    Iteration    Porosity
        Permability    Form Factor    Throat Radii Mode\n');
    %Writes output data into complete output file
    for q = 1:outputCount
        fprintf(fidOutput, '%d %d %f %f %f %e\n',outputData(q,1),
            outputData(q,2),outputData(q,3),outputData(q,4),
            outputData(q,5),outputData(q,6));
    end
    fclose('all');


    %% File Clean up
    %Copies needed files over to results section
    copyfile('*.txt',strcat(testFolder,'\Results'));
    copyfile('*Input_1.dat',strcat(testFolder,'\Results'));
    %Deletes all the unneed files
    delete *
```

```
    %Goes back to script folder to start next test
    cd(folderLoc);

    %Clears output variables for next test
    drainData = [];
    outputData = nan(100,6);
    outputCount = 0;


end


%% End of script
toc

%% Functions

%Netgen Writing
function netgenWrite(fileName,line1, latticeSize, line3, line4,
    line5, line6, line7, line8, line9, line10, line11)

%Opens netgen input file
fid = fopen(strcat(fileName,'.dat'), 'w');

%Prints data into netgenfile
fprintf(fid, '%s\n',line1);
fprintf(fid, '%i %i %i\n',latticeSize, latticeSize, latticeSize);
```

```
fprintf(fid, '%f %f %f %f\n', line3(1),line3(2),line3(3),line3(4));
fprintf(fid, '%f %f %f %f\n', line4(1),line4(2),line4(3),line4(4));
fprintf(fid, '%f %f %f %f\n', line5(1), line5(2), line5(3), line5(4));
fprintf(fid, '%f %f %f %f\n', line6(1), line6(2), line6(3), line6(4));
fprintf(fid, '%f %f\n',line7(1), line7(2));
fprintf(fid, '%f %f\n',line8(1), line8(2));
fprintf(fid, '%f\n',line9);
fprintf(fid, '%f\n',line10);
fprintf(fid, '%s\n',line11);
%closes all open files
fclose('all');
end

%Poreflow Writing
function poreflowWrite(testNum,line1, line2, line3, line4, line5, line6, line7, line8, line9)

%Opens netgen input file that is missing the last two parameters, title and
%network files
fid = fopen(strcat('poreflowInput_',testNum,'.dat'), 'a');

%Prints data into poreflow file
```

```
%Wettability
fprintf(fid, '\n\n%s\n', line1);
fprintf(fid, '%f %f %f %f\n', line2(1),line2(2),line2(3),line2(4))
    ;
fprintf(fid, '%s', line3);
%Network
fprintf(fid, '\n\n%s\n', line4);
fprintf(fid, 'F %s\n', line5);
fprintf(fid, '%s\n\n', line6);
%Title
fprintf(fid, '%s\n', line7);
fprintf(fid, '%s\n', line8);
fprintf(fid, '%s\n', line9);

fclose('all');
end

%Drain Data Import
function [drainData] = drainDataImport(filename)

%filename = 'poreTest6_04811_2_draincycle_1.csv';
fid = fopen(filename);

%Checks to make sure the filename is correct, if not it does not
    add any
%data to the inputed variable, and then ends the function
if fid == -1
```

```
    return
end

%Imports drain data from .csv file
temp = cell2mat(textscan(fid, '%f%f%f%f%f', 'Delimiter', ',', 'HeaderLines', 2));
%Adds imported drain data into the inputed variabe
drainData = [temp(:,1), temp(:,2), temp(:,3), temp(:,4)];

%Closes all open files
fclose('all');

end

%Output Data Import
function [outputData] = outputDataImport(filename)%, outputData)

%Opens output file
fid = fopen(filename);

%Checks to make sure filename is correct, if it is not then it terminates
%the function
if fid == -1%
    outputData = outputData;
    return
end
```

```
%Imports porosity [-], absolute permeability [mD], and formation factor [-]
temp = textscan(fid, '%s', 18, 'HeaderLines', 58);
%Saves porosity, permeability, and formation factor to inputed variable
%outputData = [outputData; str2double(temp{1}(3)), str2double(temp{1}(11)), str2double(temp{1}(18))];
outputData = [str2double(temp{1}(3)), str2double(temp{1}(11)), str2double(temp{1}(18))];
%Closes all open files
fclose('all');

end

%Throat Radius Import
function throatVar1 = throatRadiImport(filename)

%Opens File
fid = fopen(filename);

%Checks to make sure the filename is correct, if not it does not add any
%data to the inputed variable, and then ends the function
if fid == -1
    return
end
```

```
%Imports pore radi and pore shape factor from node file
poreData = cell2mat(textscan(fid, '%f %f %f %f %f %f', '
    HeaderLines',1));
%Throat Length
throatVar1 = poreData(:,4);


%Closes the file
fclose('all');

end
```

Random REV Plots

```
% Creates
% Script created on 9 March 2021
% Script created by Brandon Yokeley
% Script updated on 9 March 2021
%
%
% Plots ten reandomly selected networks over a range of lattice sizes to
% illustrate that REV is determined to be between a lattice size of 60 and
% 65. It also plots Sw - krw to show how water realtative
% permeability changes by lattice size. Lastly, it shows our water relative
% permeability fit for the data described in Lindquist et al, and Arns, et
% al.
%
%
% Clears all varaibles, closes all plots, and starts a timer
clear
close all
tic

%% Initial Parameters

```

```
%Plotting colors for REV Plots
lineColor = ['#E20000'; '#9B1010'; '#f58231'; '#ffe119'; '#99D649'
    ; ...
     '#1D9128'; '#013220'; '#42d4f4'; '#455DA0'; '#02075d'; ...
     '#791F89'; '#FF1AAD'; '#ff007f'; '#a9a9a9'; '#000000'];

%Location for data
folderLoc = 'C:\Users\dudem\OneDrive - Kansas State University\
    poreNetworkModel\rockTypingProject\rockData\rockTypingData\';

%Creates empty matrix for plotting data
avgData = nan(7,4);
%Counter for plotting averaged data
count = 0;
%Counter for line color
lineCounter = 0;

%% Data Plotting - REV Pltos

%Picks random test numbers
testSelection = round(240*rand(10,1));

%Loops through test data to plot REV plots
for i = testSelection'

    %Creates the working directory for this test
    testNum = strcat('test',num2str(i));
```

```
    testFolder = strcat(folderLoc,testNum,'\Results\');

    [singlePhaseData, drainData] = resultsDataImport(strcat(
        testFolder,testNum,'_Output.txt'), strcat(testFolder,
        testNum,'_Drain.txt') );

    %Calculates lattice sizes
    latSizes = unique(singlePhaseData(:,2));

    for j = [10,20,30,40,50,60,65]
        count = count + 1;
        avgData(count,:) = mean(singlePhaseData(singlePhaseData
            (:,2) == j, 2:5));

        if i == testSelection(end)

            fig2 = figure(2);
            hold on
            plot(drainData(drainData(:,1) == j, 3), drainData(
                drainData(:,1) == j, 5), '.', ...
                 'Color', lineColor(count,:), 'MarkerSize', 12)
            xlabel('S_w')
            ylabel('k_{rw}')
            set(gca, 'FontSize', 14)
            lgd = legend({'10', '20', '30', '40', '50', '60', '65'
                }, 'location', 'best', 'NumColumns',2);
            title(lgd, 'Lattice Size')
```

```
        end

    end

    %Converts permeability from mD to m^2
    avgData(:,3) = avgData(:,3) .* 9.869233e-16;

    %Counter for line color
    lineCounter = lineCounter + 1;
    %--Plotting Code
    fig1 = figure(1);
    %--Lattice Size vs Porosity
    subplot(3,1,1)
    hold on
    plot(avgData(:,1), avgData(:,2), '-', 'Color', lineColor(
        lineCounter,:), 'LineWidth', 1.25)
    xlim([0,70])
    xlabel('Lattice Size')
    ylim([0.05, 0.35])
    ylabel('Porosity')
    %--Lattice Size vs Permeability
    subplot(3,1,2)
    hold on
    plot(avgData(:,1), avgData(:,3), '-', 'Color', lineColor(
        lineCounter,:), 'LineWidth', 1.25)
    xlim([0,70])
    ylim([10e-16, 10e-10])
```

```
    yticks([10e-16, 10e-14, 10e-12, 10e-10]);
    yticklabels({'10e-16', '10e-14', '10e-12', '10e-10'})
    xlabel('Lattice Size')
    ylabel('Permeability [m^2]')
    set(gca, 'YScale', 'log')
    %--Lattice Size vs Formation Factor
    subplot(3,1,3)
    hold on
    plot(avgData(:,1), avgData(:,4), '-', 'Color', lineColor(
        lineCounter,:), 'LineWidth', 1.25)
    xlim([0,70])
    ylim([5, 2*10^3])
    yticks([5, 10^1, 10^2, 10^3, 2*10^3]);
    yticklabels({'5', '10', '100', '1000', '2000'})
    xlabel('Lattice Size')
    ylabel('Formation Factor')
    set(gca, 'YScale', 'log')




    %Resets plotting counter
    count = 0;
end

%% Data Plotting - Lindquist/Arns Data

```

```
%Imports the pore network data
twoPhaseFID = fopen('totWaterData.txt');
%Reads in synthetic network data
networkData = cell2mat(textscan(twoPhaseFID, '%f %f %f', '
    HeaderLines',1));

%Loads in digitized data from literature
lindquistData = load('lindquistData.txt');

for i = 241:244

    %Synthetic Network data
    synData = networkData(networkData(:,1) == i, :);
    %Literature Data
    expData = lindquistData(lindquistData(:,1) == i, :);
    %Converts literature data to fractional
    expData = expData ./ 100;
    %Counter for subplot
    count = count + 1;

    %--Plotting Code

    %Sets porosity of network
    lindTitles = [7.5, 13, 15, 22];
    %Sets title for subplots
    plotTitle = sprintf('Lindquist %d, %2.1f phi', count,
        lindTitles(count));
```

```
    fig3 = figure(3);
    subplot(2,2,count)
    hold on
    plot(expData(:,2), expData(:,3), '.k', 'MarkerSize', 12)
    plot(synData(:,2), synData(:,3), '-r', 'LineWidth', 1.25)
    xlabel('S_w')
    ylabel('k_{rw}')
    title(plotTitle)

end


%% Functions

function [outputVar, drainVar] = resultsDataImport(outputFile,
    drainFile)

%Sets filename
outFID = fopen(outputFile);
drainFID = fopen(drainFile);

%Checks to make sure the filename is correct, if not it does not
    add any
%data to the inputed variable, and then ends the function
if outFID == -1  || drainFID == -1
    return
```

```
end

%Imports data from .txt file
outputVar = cell2mat(textscan(outFID, '%f %f %f %f %f %f', '
    HeaderLines',2));
drainVar = cell2mat(textscan(drainFID, '%f %f %f %f %f %f', '
    HeaderLines',2));
fclose('all');

end
```

Initial processing script

```
% Tests rock typing technique on six test pore networks
% Script created on 31 Janurary 2020
% Script created by Brandon Yokeley
% Script updated on 1 October 2020
%
% Clears all varaibles, closes all plots, and starts a timer
clear
close all
tic

%% Data import Setup

%Lattice Size to import
latSize = 65;
%Rock type for plotting
plottingColor = load('plottingColors.csv');
%Number of iterations
itNum = 10;
%Folder locations to store plots
%Laptop location
folderLoc = '/Users/yokeleyba/OneDrive - Kansas State University/
    poreNetworkModel/rockTypingProject/rockData/rockTypingData/';
%Personal Desktop location
%folderLoc = 'C:\Users\dudem\OneDrive - Kansas State University\
    poreNetworkModel\rockTypingProject\rockData\rockTypingData\';
```

```
%Lab Desktop location
%folderLoc = 'C:\Users\yokeleyba\OneDrive - Kansas State University\poreNetworkModel\rockTypingProject\rockData\rockTypingData\';
oneDrive = 'C:\Users\yokeleyba\OneDrive - Kansas State University\poreNetworkModel\rockTypingProject\rockData\plots';
%Loads in coordination numbers so that they can be color coded
connNumber = load('connNumber.csv');
%Test Counter
count = 0;
%Loads in colors for plotting
%plotColors = load('plottingColors.csv');
alpha3 = 0;
alpha4 = 0;
alphaValues = [3.021579832, 2.867739711, 2.852532311, 2.916683901, 2.877303371, 2.852532311];

lineColor = ['#E20000'; '#f58231'; '#ffe119'; '#99D649'; '#1D9128'; '#42d4f4'; '#455DA0'; '#791F89'; '#FF1AAD'; '#a9a9a9'; '#000000'; '#9B1010'];

porosity = nan(1,240);
formFactor = nan(1,240);
permeability = nan(1,240);
kMeansData = [];
kMeansOutputData = [];
rcSwData = cell(240,1);
```

```
seData = cell(240,1);
sampleData = nan(100,240);
sampleData_2 = [];
krwData = nan(100,240);
%sampleData_2 = cell(240,1);

%% Data import

%Loops through samples
for i = 1:240
    count = count+1;

    %Creates the working directory for this test
    testNum = strcat('test',num2str(count));
    testFolder = strcat(folderLoc,testNum,'/Results/');

    %Imports absolute permeability, formation factor, porosity,
        and all
    %drain data (water saturation, Kro, Krw, and cappilary
        pressure)
    [outputData, drainData] = resultsDataImport(strcat(testFolder,
        testNum,'_Output.txt'), strcat(testFolder,testNum,'_Drain.
        txt'));

    %Sorts drain data to extract REV value for respective pore
        network
    drainData = drainData(drainData(:,1) == latSize, 2:6);
```

```
%Sorts output data to extract REV value for respective pore network
outputData = outputData(outputData(:,2) == latSize, 2:6);
%Creates empty cell array to store simulated water saturation into
simSw = cell(itNum,1);
%Creates empty cell array to store simulated water relative
%permeability into
simKrw = cell(itNum,1);
%Creates empty cell array to store simulated cappilary pressure curves
simCp = cell(itNum,1);
%Creates an empty matrix to store the length of each simulated sw-krw
%curve into
dataLength = nan(itNum,1);

%Bins drain data to extract only one Krw curve...mikama or spline
%interpolation functions
%--Makima interpolation
%Simulated sw values for modified akima piecewise cubic Hermite
%interpolation
for j = 1:itNum
    %Filters out iteration number
    tempDrain = drainData(drainData(:,1) == j, 2:5);
```

```
        %Finds water saturation values that repeats
        [~,A,~] = unique(tempDrain(:,1));
        %Filters out repeated water saturation values
        tempDrain = tempDrain(A,:);
        %Finds Krw repeated values
        [~,A,~] = unique(tempDrain(:,3));
        %Filters out repeated Krw values
        tempDrain = tempDrain(A,:);
        %Finds repeated Cap Pressure Values
        [~,A,~] = unique(tempDrain(:,2));
        %Filters out repeated Cap Pressure Values
        tempDrain = tempDrain(A,:);
        %Creates equally spaced water saturation values that will
        %correspond to the interpolated Krw values
        simSw{j} = linspace(min(tempDrain(:,1)),max(tempDrain(:,1)
            ),100);
        %simSw{j} = linspace(min(tempDrain(:,1)),max(tempDrain
            (:,1)),length(tempDrain(:,1)));
        %Interpolates Krw values
        simKrw{j} = makima(tempDrain(:,1), tempDrain(:,3), simSw{j
            });
        %Interpolates Cap Pressure Values
        simCp{j} = makima(tempDrain(:,1), tempDrain(:,2), simSw{j
            });
        %Inputs length of each interpolated data set
        dataLength(j) = length(simSw{j});
    end
```

```
%Creates a temperorary matrix to store simulated data into
tempSw = nan(itNum, max(dataLength));
tempKrw = nan(itNum, max(dataLength));
tempCp = nan(itNum, max(dataLength));

%Adds simulated data into newly created matrix so that the simulated
%data can then be averaged
for j = 1:itNum
    tempSw(j,1:length(simSw{j})) = simSw{j};
    tempKrw(j,1:length(simKrw{j})) = simKrw{j};
    tempCp(j,1:length(simCp{j})) = simCp{j};
end

%Converts cell array to matrix array so that the data can be averaged
simSw = mean(tempSw);
simKrw = mean(tempKrw);
simCp = mean(tempCp);
%Uncomment this if there is only one iteration
%simSw = tempDrain(:,1);
%simKrw = tempDrain(:,3);

%figure(6)
%plot(drainData(:,2), drainData(:,4), 'k.')
%hold on
```

```
%ylabel('Water Relative Permeability', 'FontSize', 16')
%xlabel('Water Saturation', 'FontSize', 16)
%set(gca, 'YScale', 'log', 'XScale', 'log')

%Averages water saturation values for this rock sample
%simSw = mean(simSw, 'omitnan');
%Averages interpolated Krw values
%simKrw = mean(simKrw, 'omitnan');
%Averages output data for extracted REV value
outputData = mean(outputData(:,2:5), 1);
%Converts permeability from milidarcy to m^2
permeability(i) = outputData(2);
outputData(2) = outputData(2) * 9.869233e-16;
%Finds calculated absolute permeability value to then determine alpha
%valueq
calcK_3 = abs(outputData(4)^2 / ((72.2/4)*outputData(3))-outputData(2));
calcK_4 = abs(outputData(4)^2 / ((53.3/4)*outputData(3))-outputData(2));

%Determines alpha value based off the comparison between the calculated
%permeability and the actual permeabililty
if calcK_3 < calcK_4
    %Calculates rcSw
    rcSw = outputData(4)*(simKrw.^(1/3));
```

```
        %rcSw = rcSw/outputData(4);
        alpha3 = alpha3 + 1;
        %Calculates rcSw to plot versus cap pressure
        %rcSw = simKrw.^(1/3);

    end

    if calcK_3 > calcK_4
        %Calculates rcSw
        rcSw = outputData(4)*(simKrw.^(1/4));
        %rcSw = rcSw/outputData(4);
        alpha4 = alpha4 + 1;
        %Calculates rcSw to plot versus cap pressure
        %rcSw = simKrw.^(1/4);
    end

    %Divides by maximum pore throat radi
    %     if (count < 121 && count > 60) || (count < 241 && count
       > 180)
    %         rcSw = rcSw / 100;
    %     else
    %         rcSw = rcSw / 10;
    %     end


    %--Calculates effective water saturation
    sE = (simSw - min(simSw))/(1-min(simSw));
```

```
%--Normalizes Cap Pressure
%cP = simCp / max(simCp);
cP = (simCp - min(simCp))/(max(simCp) - min(simCp));

%--Saves NON normalized data to be clustered and plotted



%--Normalizes rcSW
%rcSw = rcSw / max(rcSw);
rcSw = (rcSw - min(rcSw))/(max(rcSw) - min(rcSw));
%rcSw_Calc = outputData(4)*(simKrw.^(1/alphaValues(i)));


%--Saves normalized data for plotting
%Used for plotting
sampleData(1:100,i) = rcSw;
sampleData(101:200,i) = sE;
%Used for curve fitting
rcSwData{i} = rcSw';
seData{i} = sE';
krwData(:,i) = simKrw;

%Sets line color for plotting
curveColor = string(lineColor(plottingColor(i),:));

%plots rc(Sw) vs effective water saturation
```

```
    fig1 = figure(1);
    %sgtitle({'Test Number: ', num2str(count)})

    %---Sw vs Krw
    subplot(1,2,1)
    hold on
    plot(simSw, simKrw, 'Color', curveColor, 'LineWidth', 1.25)
    xlabel('r_c(S_w)', 'FontSize', 16)
    ylabel('S_e', 'FontSize', 16)

    subplot(1,2,2)
    hold on
    plot(rcSw, sE, 'Color', curveColor, 'LineWidth', 1.25)
    plot(rcSw_Calc, sE, 'b', 'LineWidth',1.25)
    xlabel('r_c(S_w)', 'FontSize', 16)
    ylabel('S_e', 'FontSize', 16)
    set(gca, 'YScale', 'log', 'XScale', 'log')


    %---rc(Sw) vs sE
    subplot(1,2,1)
    hold on
    plot(rcSw, sE, 'k', 'LineWidth', 1.25)
    xlabel('r_c(S_w)', 'FontSize', 16)
    ylabel('S_e', 'FontSize', 16)

    subplot(1,2,2)
```

```
    hold on
    plot(rcSw, sE, 'k', 'LineWidth', 1.25)
    plot(rcSw_Calc, sE, 'b', 'LineWidth',1.25)
    xlabel('r_c(S_w)', 'FontSize', 16)
    ylabel('S_e', 'FontSize', 16)
    set(gca, 'YScale', 'log', 'XScale', 'log')

%       figure(2)
%       hold on
%       plot(simSw, simKrw, 'LineWidth', 1.25)
%     plot(simKrw, sE, 'LineWidth', 1.25)
%       plot(drainData(:,2), drainData(:,4), '.k')
%       xlabel('Sw', 'FontSize', 16)
%       ylabel('K_rw', 'FontSize', 16)
%     set(gca, 'XScale', 'log', 'YScale', 'log')
    %ylim([1e-4, 1])
    %
%     figure(3)
%     subplot(1,2,1)
%     hold on
%     plot(rcSw, cP, 'k', 'LineWidth', 1.25)
%     xlabel('r_c(S_w)', 'FontSize', 16)
%     ylabel('P_c [Pa]' , 'FontSize', 16)
%     subplot(1,2,2)
%     hold on
%     plot(rcSw, cP, 'k', 'LineWidth', 1.25)
%     xlabel('r_c(S_w)', 'FontSize', 16)
```

```
%       ylabel('P_c [Pa]' , 'FontSize', 16)
%       set(gca, 'YScale', 'log', 'XScale', 'log')

    porosity(i) = outputData(1);
    formFactor(i) = outputData(3);
    %Adds rcSw and Se into k-means matrix so that k-means can be calculated
    %kMeansData = [kMeansData; rcSw', sE'];
    %kMeansOutputData = [kMeansOutputData; outputData];
    %Curve clustering....or trying to
    %rcSwData(1:max(dataLength),i) = log(rcSw);
    %seData(1:max(dataLength),i) = sE';
    %sampleData_2{i} = [log(rcSw), log(sE)];
end

%% CCT Toolbox

%--Must set path for CCT toolbox before it can be used.

%Sets X data for curve cluster toolbox
trajs.X = rcSwData;
%Sets Y data for curve cluster toolbox
trajs.Y = seData;
%Sets the clustering method
ops.method = 'lrm_b';
%Sets dimensions of data, in this case two.
ops.order = 3;
```

```
%Sets the number of clusters
ops.K = 11;
%Sets normalization
ops.zero = 'none';
%Sets number of iterations
%ops.NumEMStarts = 3;
ops.TrainLhood = 50;
%Clusters data
model = curve_clust(trajs,ops);
%% Plotting Code

%Sets cluster numbe to bottom of sample data
sampleData(201,:) = model.C';
manClusters = load('clusters.csv');

%----------Automated Clustering----------
%---For loop to plotted clustered data
for i = 1:240
    figure(1)
%     subplot(1,2,1)
%     hold on
%     plot(sampleData(1:100,i), sampleData(101:200,i), 'Color',
    lineColor(model.C(i),:))
%     set(gca, 'YScale', 'log', 'XScale', 'log')
%     xlabel('r_c(S_w)', 'FontSize', 16)
%     ylabel('S_e', 'FontSize', 16)

```

```
%       subplot(1,2,2)
    hold on
    plot(sampleData(1:100,i), sampleData(101:200,i), 'Color',
        lineColor(model.C(i),:))
    xlabel('r_c(S_w)', 'FontSize', 16)
    ylabel('S_e', 'FontSize', 16)
    title('Automated Clustering', 'FontSize', 24)


end

%---Indvidual Clustered Curves
for i = 1:ops.K
    figure(2)
    sgtitle({'Automated Clustering', 'Indvidual Clustered Curves'
        }, 'FontSize', 24)
    subplot(3,4,i)
    plot(sampleData(1:100, sampleData(201,:)==i), sampleData
        (101:200, sampleData(201,:)==i), 'Color',lineColor(i,:))
    %plot(krwData(:, sampleData(201,:)==i), sampleData(101:200,
        sampleData(201,:)==i), 'Color',lineColor(i,:))
    xlabel('rcSw')
    ylabel('sE')
    indPlotTitle = sprintf('Cluster %i', i);
    title(indPlotTitle)
end

```

```
%---Porosity versus Absolute Permeabiltiy
for i = 1:ops.K
    figure(3)
    sgtitle({'Automated Clustering', 'Porosity versus Absolute
        Permeabiltiy'}, 'FontSize', 24)
    subplot(3,4,i)
    plot(porosity(sampleData(201,:)==i), permeability(sampleData
        (201,:)==i), '.', 'Color',lineColor(i,:), 'MarkerSize', 12)
    xlim([0.05 0.35])
    ylim([0 10e4])
    set(gca, 'YScale', 'log')
    xlabel('Porosity')
    ylabel('Permeability [mD]')
    indPlotTitle = sprintf('Cluster %i', i);
    title(indPlotTitle)
end


%---Absolute Permeability versus Formation Factor
for i = 1:ops.K
    figure(4)
    sgtitle({'Automated Clustering', 'Absolute Permeability versus
        Formation Factor'}, 'FontSize', 24)
    subplot(3,4,i)
    plot(1./formFactor(sampleData(201,:)==i), permeability(
        sampleData(201,:)==i), '.', 'Color',lineColor(i,:), '
        MarkerSize', 12)
```

```
    %set(gca, 'YScale', 'log')
    ylabel('Permeability [mD]')
    xlabel('1/F')
    indPlotTitle = sprintf('Cluster %i', i);
    title(indPlotTitle)
end

%----------Manually Fitted Curves----------
sampleData(201,:) = manClusters';
%---For loop to plotted clustered data
for i = 1:240
    figure(5)
%     subplot(1,2,1)
%     hold on
%     plot(sampleData(1:100,i), sampleData(101:200,i), 'Color',
    lineColor(model.C(i),:))
%     set(gca, 'YScale', 'log', 'XScale', 'log')
%     xlabel('r_c(S_w)', 'FontSize', 16)
%     ylabel('S_e', 'FontSize', 16)

%     subplot(1,2,2)
    hold on
    plot(sampleData(1:100,i), sampleData(101:200,i), 'Color',
        lineColor(manClusters(i),:))
    xlabel('r_c(S_w)', 'FontSize', 16)
    ylabel('S_e', 'FontSize', 16)
    title('Manually Fitted', 'FontSize', 24)
```

```
end

%---Indvidual Clustered Curves
for i = 1:max(manClusters)
    figure(6)
    sgtitle({'Manually Fitted', 'Manually Fitted Indvidual Clustered Curves'}, 'FontSize', 24)
    subplot(3,4,i)
    plot(sampleData(1:100, sampleData(201,:)==i), sampleData(101:200, sampleData(201,:)==i), 'Color',lineColor(i,:))
    %plot(krwData(:, sampleData(201,:)==i), sampleData(101:200, sampleData(201,:)==i), 'Color',lineColor(i,:))
    xlabel('rcSw')
    ylabel('sE')
    indPlotTitle = sprintf('Cluster %i', i);
    title(indPlotTitle)
end

%---Porosity versus Absolute Permeabiltiy
for i = 1:max(manClusters)
    figure(7)
    sgtitle({'Manually Fitted','Porosity versus Absolute Permeabiltiy'}, 'FontSize', 24)
    subplot(3,4,i)
    plot(porosity(sampleData(201,:)==i), permeability(sampleData(201,:)==i), '.', 'Color',lineColor(i,:), 'MarkerSize', 12)
```

```
    xlim([0.05 0.35])
    ylim([0 10e4])
    set(gca, 'YScale', 'log')
    xlabel('Porosity')
    ylabel('Permeability [mD]')
    indPlotTitle = sprintf('Cluster %i', i);
    title(indPlotTitle)
end


%---Absolute Permeability versus Formation Factor
for i = 1:max(manClusters)
    figure(8)
    sgtitle({'Manually Fitted', 'Absolute Permeability versus
        Formation Factor'}, 'FontSize', 24)
    subplot(3,4,i)
    plot(1./formFactor(sampleData(201,:)==i), permeability(
        sampleData(201,:)==i), '.', 'Color',lineColor(i,:), '
        MarkerSize', 12)
    %set(gca, 'YScale', 'log')
    ylabel('Permeability [mD]')
    xlabel('1/F')
    indPlotTitle = sprintf('Cluster %i', i);
    title(indPlotTitle)
end

%% Reresentative Curve Plotting
```

```
%---Plots Manually picked representative curves
for i = 1:max(manClusters)

    %---Averaging code
    %Creates prerepresentative rcSw vs Se Curves
    repCurve = [mean(sampleData(1:100, sampleData(201,:)==i),2),
        mean(sampleData(101:200, sampleData(201,:)==i),2)];
    %Creates average porosity X cluster
    repPorosity = mean(porosity(sampleData(201,:)==i));
    %Creates average absoulute permeability for X cluster
    repPermeability = mean(permeability(sampleData(201,:)==i));
    %Creates average formation factor for X cluster
    repForm = mean(1./formFactor(sampleData(201,:)==i));

    %---Plots data
    figure(9)
    sgtitle('Manually Fitted Representative Curves', 'FontSize',
        24)
    %Representative Curves
    subplot(3,2,[1,3,5])
    plot(repCurve(:,1), repCurve(:,2), 'Color',lineColor(i,:), '
        LineWidth',1.25)
    hold on
    xlabel('r_c(S_w)', 'FontSize', 16)
    ylabel('S_e', 'FontSize', 16)

```

```
    %Porosity vs Absolute Permeability
    subplot(3,2,2)
    hold on
    plot(repPorosity, repPermeability, '.', 'MarkerSize',20, 'Color',lineColor(i,:))
    xlim([0.05 0.35])
    set(gca, 'XScale', 'log', 'YScale', 'log')
    xlabel('Porosity')
    ylabel('Permeability [mD]')

    %Formation Factor vs Absolute Permeability
    subplot(3,2,4)
    hold on
    plot(repForm, repPermeability, '.', 'MarkerSize',20, 'Color', lineColor(i,:))
    set(gca, 'XScale', 'log', 'YScale', 'log')
    xlabel('1/F')
    ylabel('Permeability [mD]')

end

%%

cluster1 = [sampleData(1:100, sampleData(201,:)==6), sampleData(101:200, sampleData(201,:)==6)];
cluster2 = [sampleData(1:100, sampleData(201,:)==8), sampleData(101:200, sampleData(201,:)==8)];
```

```
figure(10)
hold on
plot(sampleData(1:100, sampleData(201,:)==6), sampleData(101:200,
    sampleData(201,:)==6), 'Color',lineColor(1,:), 'LineWidth'
    ,0.25)
plot(sampleData(1:100, sampleData(201,:)==8), sampleData(101:200,
    sampleData(201,:)==8), 'Color',lineColor(4,:), 'LineWidth'
    ,0.25)
%plot(cluster1(:,1), cluster1(:,2), 'Color',lineColor(1,:), '
    LineWidth',1.25)
%plot(cluster2(:,1), cluster2(:,2), 'Color',lineColor(4,:), '
    LineWidth',1.25)
xlabel('r_c(S_w)', 'FontSize', 16)
ylabel('S_e', 'FontSize', 16)
title('Manually Fitted', 'FontSize', 24)


toc



%% Extra Code
%
%
% %Uses k-means clustering to attempt to group like rock types
    together
```

```
% %kMeansData = kMeansData(~isnan(kMeansData(:,1)), :);
% %[idx, C] = kmeans(kMeansData(:,1), 7, 'Replicates', 10);
% %[idx2, C2] = kmeans(kMeansOutputData(:,2)/kMeansOutputData(:,3)
    , 7, 'Replicates', 10);
%
% %Saves data so that it alike curves can be clustered together
% %totData = rcSwData;
% %sampleData = nan(264,1);
% %sampleData(1:max(dataLength),i) = rcSw;
% %sampleData(133:132+max(dataLength),i) = sE;
% %totData(isnan(totData)) = 0;
% sampleData(isnan(sampleData)) = 0;
% sampleData(sampleData == -Inf) = 1;
% %totData(totData == -Inf) = 0;
%
%
% %Summation matrix
% dataDiff = nan(240,240);
% %For loop for curve clustering. Will sum difference differences
    in curves
% %to find similar curves
% for i = 1:240
%     dataDiff(i,:) = abs(sum(sampleData- sampleData(:,i)));
% end
%
% %Finds all curves that have a summation difference less than the
    stated
```

```
% %value
% dataDiff(dataDiff > 7) = 0;
% dataDiff(dataDiff < 7 & dataDiff ~= 0) = 1;
% %Deletes all the lower diagonal values to find a like curves
% dataDiff = tril(dataDiff);
% %Finds row/col of each of a like curves
% [row, col] = find(dataDiff == 1);
%
% totData = [col,row];
%
% %Finds alike curves an groups them together
% curveClusters = [];
% for i = 1:240
%     curveClusters = [curveClusters; totData(totData(:,1) == i,
    :)];
% end
%
% %Finds unique clusters and sorts them to find groups
% [~, A, ~] = unique(curveClusters(:,2));
% curveClusters = curveClusters(A,:);
% curveClusters = sortrows(curveClusters);
%
% %Sums curves that are similair to find how many curves are
    similar to that
% %one
% B = sum(dataDiff);
%
```

```
%
%%Title for rocktyping plot
%%title('Samples 1 - 240', 'FontSize',24)
%%Legend for rock typing plot when seperating by coordination number
%%legend([p1, p2, p3], {'Cord Num: 2', 'Cord Num: 4', 'Cord: Num 6'}, 'FontSize', 16, 'Location', 'eastoutside')
%
%% figure(4)
%% subplot(1,3,1)
%% histogram(porosity,10)
%% xlabel('Porosity', 'FontSize',16)
%% subplot(1,3,2)
%% histogram(permeability,10)
%% xlabel('Permeability [mD]', 'FontSize',16)
%% subplot(1,3,3)
%% histogram(formFactor,10)
%% xlabel('Formation Factor', 'FontSize',16)
%
%%print(fig1, 'rockTypingPlot', '-dpng', '-r300')
%
%% figure(5)
%% gscatter(kMeansData(:,1), kMeansData(:,2), idx)
%% xlabel('r_c(S_w)', 'FontSize', 16)
%% ylabel('S_e', 'FontSize', 16)
%% set(gca, 'YScale', 'log', 'XScale', 'log')
%%
```

```
%% figure(6)
%% gscatter(1/kMeansOutputData(:,3), kMeansOutputData(:,2), idx2)
%% plot(kMeansOutputData(:,1), kMeansOutputData(:,2), 'k.')
%% xlabel('Porosity', 'FontSize', 18)
%% ylabel('Absolute Permeability', 'FontSize', 18)
%% set(gca, 'YScale', 'log')
%
%
```

Final processing script

```
% Conducts rock typing technique on complete dataset
% Script created on 9 December 2020
% Script created by Brandon Yokeley
% Script updated on 9 March 2021
%
% The final processing of all our data. This loads in pre averaged
% networks, and processes them according to our rock typing procedure.
% Additionally, it also clusters the networks ussing the CCToolbox, matches
% our data to methods in the literature. And creates the majority of the
% plots within the results section of our thesis.
%
% Clears all varaibles, closes all plots, and starts a timer
clear
close all
```

```
tic

%% Initial Parameters

%Plotting colors for represetative rock types
lineColor = ['#E20000'; '#9B1010'; '#f58231'; '#ffe119'; '#99D649'
    ; ...
     '#1D9128'; '#013220'; '#42d4f4'; '#455DA0'; '#02075d'; ...
     '#791F89'; '#FF1AAD'; '#ff007f'; '#a9a9a9'; '#000000'];

%Initial Parameter setup for data processing

%--Plotting Initial Parmaeters
sampleData = nan(100,246);
sampleData_2 = nan(100,246);
totSw = nan(100,246);
totKrw = nan(100,246);
%--Curve Fitting Initial Parameters
rcSwData = cell(246,1);
seData = cell(246,1);
temData = cell(246,1);
swData = cell(246,1);

%Imports the pore network data
singlePhaseFID = fopen('totSinglePhaseData.txt');
twoPhaseFID = fopen('totWaterData.txt');

```

```
singlePhaseData = cell2mat(textscan(singlePhaseFID, '%f %f %f %f %f', 'HeaderLines',1));
twoPhaseData = cell2mat(textscan(twoPhaseFID, '%f %f %f', 'HeaderLines',1));

%Set between 0, 1, and 2 to use curve curve clustering toolbox
%---0 Do not use CCT
%---1 Use CCT for Two-Phae rock typing
%---2 Use CCT for TEM-Function
autoCluster = 0;


%% Data Processing

close all

%Sets empty vector for irreducible water saturation
swi = nan(length(singlePhaseData),1);

for i = 1:length(singlePhaseData)

    %Finds calculated absolute permeability value to then determine alpha
    %valueq
    calcK_3 = abs(singlePhaseData(i,5)^2 / ((72.2/4)*singlePhaseData(i,4))-singlePhaseData(i,3));
    calcK_4 = abs(singlePhaseData(i,5)^2 / ((53.3/4)*
```

```
singlePhaseData(i,4))-singlePhaseData(i,3));

%Parses out water saturation data and water relative permeability data
simSw = twoPhaseData(twoPhaseData(:,1) == i, :);
simKrw = simSw(:,3);
simSw = simSw(:,2);

%Calculates irreducible water saturation
swi(i) = min(simSw);

%--Caluclates critical pore radii at varying water saturations
%Determines alpha value based off the comparison between the calculated
%permeability and the actual permeabililty
if calcK_3 < calcK_4
    %Calculates rcSw
    rcSw = singlePhaseData(i,5)*(simKrw.^(1/3));
end

if calcK_3 > calcK_4
    %Calculates rcSw
    rcSw = singlePhaseData(i,5)*(simKrw.^(1/4));
end

%--Calculates effective water saturation
```

```
    sE = (simSw - min(simSw))/(1-min(simSw));

    %--Normalizes rcSW
    rcSw = rcSw / max(rcSw);

    %--Saves normalized data for plotting
    %--Plotting Data
    totSw(:,i) = simSw;
    totKrw(:,i) = simKrw;
    sampleData(1:100,i) = rcSw;
    sampleData(101:200,i) = sE;

    %--Curve Fit Data
    rcSwData{i} = rcSw;
    seData{i} = sE;
    %-TEM-Function [d/cp]
    temData{i} = (singlePhaseData(i,3) .* simKrw) ./
        singlePhaseData(i,2) .* 1013249965828.1448;
    swData{i} = simSw;
    sampleData_2(1:100,i) = temData{i};
    sampleData_2(101:200,i) = simSw;

    %--Plotting Code - Complete data set
    %           Plots two-phase data
    %           fig1 = figure(1);
    %           hold on
    %           plot(simSw, simKrw, 'k', 'LineWidth', 0.5);
```

```
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig2 = figure(2);
%           hold on
%           plot(simSw, simKrw, 'k', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([10^-5,1])
%           ylim([10e-12,1])
%           xticks([10^-5, 10^-4, 10^-3, 10^-2, 10^-1, 10^0]);
%           xticklabels({'10^{-5}', '10^{-4}', '10^{-3}',
    '10^{-2}', '10^{-1}', '10^0'})
%           xtickangle(30)
%           yticks([10^-12, 10^-10, 10^-8, 10^-6, 10^-4, 10^-2,
    10^0])
%           yticklabels({'10^{-12}', '10^{-10}', '10^{-8}',
    '10^{-6}', '10^{-4}', '10^{-2}', '10^0'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', '
    log')
%
%
%           fig3 = figure(3);
%           hold on
```

```
%           plot(sE, rcSw, 'k', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig4 = figure(4);
%           hold on
%           plot(sE, rcSw, 'k', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([5.5e-3,1])
%           xticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           xticklabels({'0.0055', '0.01','0.1', '1'})
%           xtickangle(30)
%           ylim([5.5e-3,1])
%           yticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           yticklabels({'0.0055', '0.01','0.1', '1'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')

%     %--Plotting Code - Contact Angle
%
%     %Plots two-phase data
%      if i <121 || i > 240
%           fig1 = figure(1);
```

```
%           hold on
%           plot(simSw, simKrw, 'k', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig2 = figure(2);
%           hold on
%           plot(simSw, simKrw, 'k', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([10^-5,1])
%           ylim([10e-12,1])
%           xticks([10^-5, 10^-4, 10^-3, 10^-2, 10^-1, 10^0]);
%           xticklabels({'10^{-5}', '10^{-4}', '10^{-3}',
    '10^{-2}', '10^{-1}', '10^0'})
%           xtickangle(30)
%           yticks([10^-12, 10^-10, 10^-8, 10^-6, 10^-4, 10^-2,
    10^0])
%           yticklabels({'10^{-12}', '10^{-10}', '10^{-8}',
    '10^{-6}', '10^{-4}', '10^{-2}', '10^0'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', '
    log')
%
%
```

```
%           fig3 = figure(3);
%           hold on
%           plot(sE, rcSw, 'k', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig4 = figure(4);
%           hold on
%           plot(sE, rcSw, 'k', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([5.5e-3,1])
%           xticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           xticklabels({'0.0055', '0.01','0.1', '1'})
%           xtickangle(30)
%           ylim([5.5e-3,1])
%           yticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           yticklabels({'0.0055', '0.01','0.1', '1'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')
%       else
%           fig1 = figure(1);
%           hold on
%           plot(simSw, simKrw, 'r', 'LineWidth', 0.5);
```

```
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig2 = figure(2);
%           hold on
%           plot(simSw, simKrw, 'r', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([10^-5,1])
%           ylim([10e-12,1])
%           xticks([10^-5, 10^-4, 10^-3, 10^-2, 10^-1, 10^0]);
%           xticklabels({'10^{-5}', '10^{-4}', '10^{-3}',
    '10^{-2}', '10^{-1}', '10^0'})
%           xtickangle(30)
%           yticks([10^-12, 10^-10, 10^-8, 10^-6, 10^-4, 10^-2,
    10^0])
%           yticklabels({'10^{-12}', '10^{-10}', '10^{-8}',
    '10^{-6}', '10^{-4}', '10^{-2}', '10^0'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', '
    log')
%
%
%           fig3 = figure(3);
%           hold on
```

```
%           plot(sE, rcSw, 'r', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig4 = figure(4);
%           hold on
%           plot(sE, rcSw, 'r', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([5.5e-3,1])
%           xticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           xticklabels({'0.0055', '0.01','0.1', '1'})
%           xtickangle(30)
%           ylim([5.5e-3,1])
%           yticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           yticklabels({'0.0055', '0.01','0.1', '1'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')
%       end


%---Plotting Code - Pore Throat Size
%       if (i < 61) || (i > 119 && i < 181)
%           %Plots two-phase data
```

```
%           fig1 = figure(1);
%           hold on
%           plot(simSw, simKrw, 'k', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig2 = figure(2);
%           hold on
%           plot(simSw, simKrw, 'k', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([10^-5,1])
%           ylim([10e-12,1])
%           xticks([10^-5, 10^-4, 10^-3, 10^-2, 10^-1, 10^0]);
%           xticklabels({'10^{-5}', '10^{-4}', '10^{-3}',
    '10^{-2}', '10^{-1}', '10^0'})
%           xtickangle(30)
%           yticks([10^-12, 10^-10, 10^-8, 10^-6, 10^-4, 10^-2,
    10^0])
%           yticklabels({'10^{-12}', '10^{-10}', '10^{-8}',
    '10^{-6}', '10^{-4}', '10^{-2}', '10^0'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', '
    log')
%
```

```
%
%           fig3 = figure(3);
%           hold on
%           plot(sE, rcSw, 'k', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig4 = figure(4);
%           hold on
%           plot(sE, rcSw, 'k', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([5.5e-3,1])
%           xticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           xticklabels({'0.0055', '0.01','0.1', '1'})
%           xtickangle(30)
%           ylim([5.5e-3,1])
%           yticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           yticklabels({'0.0055', '0.01','0.1', '1'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')
%       elseif i > 240
%           %Plots two-phase data
%           fig1 = figure(1);
```

```
%           hold on
%           plot(simSw, simKrw, 'b', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig2 = figure(2);
%           hold on
%           plot(simSw, simKrw, 'b', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%          xlim([10^-5,1])
%           ylim([10e-12,1])
%           xticks([10^-5, 10^-4, 10^-3, 10^-2, 10^-1, 10^0]);
%           xticklabels({'10^{-5}', '10^{-4}', '10^{-3}',
    '10^{-2}', '10^{-1}', '10^0'})
%           xtickangle(30)
%           yticks([10^-12, 10^-10, 10^-8, 10^-6, 10^-4, 10^-2,
    10^0])
%           yticklabels({'10^{-12}', '10^{-10}', '10^{-8}',
    '10^{-6}', '10^{-4}', '10^{-2}', '10^0'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', '
    log')
%
%
```

```
%           fig3 = figure(3);
%           hold on
%           plot(sE, rcSw, 'b', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig4 = figure(4);
%           hold on
%           plot(sE, rcSw, 'b', 'LineWidth', 0.5);
%           xlabel('S_e', 'FontSize', 16)
%           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
%           xlim([5.5e-3,1])
%           xticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           xticklabels({'0.0055', '0.01','0.1', '1'})
%           xtickangle(30)
%           ylim([5.5e-3,1])
%           yticks([5.5e-3, 10e-3, 10e-2, 10e-1])
%           yticklabels({'0.0055', '0.01','0.1', '1'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')
%       else
%           %Plots two-phase data
%           fig1 = figure(1);
%           hold on
```

```
%           plot(simSw, simKrw, 'r', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([0,1])
%           ylim([0,1])
%           set(gca, 'FontSize',14)
%
%           fig2 = figure(2);
%           hold on
%           plot(simSw, simKrw, 'r', 'LineWidth', 0.5);
%           xlabel('S_w', 'FontSize', 16)
%           ylabel('k_{rw}', 'FontSize', 16)
%           xlim([10^-5,1])
%           ylim([10e-12,1])
%           xticks([10^-5, 10^-4, 10^-3, 10^-2, 10^-1, 10^0]);
%           xticklabels({'10^{-5}', '10^{-4}', '10^{-3}',
    '10^{-2}', '10^{-1}', '10^0'})
%           xtickangle(30)
%           yticks([10^-12, 10^-10, 10^-8, 10^-6, 10^-4, 10^-2,
    10^0])
%           yticklabels({'10^{-12}', '10^{-10}', '10^{-8}',
    '10^{-6}', '10^{-4}', '10^{-2}', '10^0'})
%           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', '
    log')
%
%
%           fig3 = figure(3);
```

```
    %           hold on
    %           plot(sE, rcSw, 'r', 'LineWidth', 0.5);
    %           xlabel('S_e', 'FontSize', 16)
    %           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
    %           xlim([0,1])
    %           ylim([0,1])
    %           set(gca, 'FontSize',14)
    %
    %           fig4 = figure(4);
    %           hold on
    %           plot(sE, rcSw, 'r', 'LineWidth', 0.5);
    %           xlabel('S_e', 'FontSize', 16)
    %           ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)
    %           xlim([5.5e-3,1])
    %           xticks([5.5e-3, 10e-3, 10e-2, 10e-1])
    %           xticklabels({'0.0055', '0.01','0.1', '1'})
    %           xtickangle(30)
    %           ylim([5.5e-3,1])
    %           yticks([5.5e-3, 10e-3, 10e-2, 10e-1])
    %           yticklabels({'0.0055', '0.01','0.1', '1'})
    %           set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')
    %       end
end


%% Automated Clustering
```

```
%Uses curve clustering to automate the selection of our data set
    using
%critical pore radii method
if autoCluster == 1


    %--Must set path for CCT toolbox before it can be used.

    %Sets X data for curve cluster toolbox
    trajs.X = rcSwData;
    %Sets Y data for curve cluster toolbox
    trajs.Y = seData;
    %Sets the clustering method
    ops.method = 'lrm_b';
    %Sets dimensions of data, in this case two.
    ops.order = 3;
    %Sets the number of clusters
    ops.K = 12;
    %Sets normalization
    ops.zero = 'none';
    %Sets number of iterations
    %ops.NumEMStarts = 3;
    ops.TrainLhood = 50;
    %Clusters data
    model = curve_clust(trajs,ops);

```

```
%Loads in manually selected curve clusters
manClusters = model.C;
%Appends cluster number to bottom of dataset
sampleData(201,:) = manClusters';

%--Indvidual Clustered Curves
for i = 1:max(manClusters)
    %-- Sw vs Krw
    %           fig4 = figure(4);
    %           subplot(3,4,i)
    %           plot(totSw(1:100, totSw(101,:)==i), totKrw
        (1:100, totKrw(101,:)==i), 'Color',lineColor(i,:))
    %           xlabel('S_w')
    %           ylabel('k_{rw}')
    %           indPlotTitle = sprintf('Rock Type %i', i);
    %           xlim([10^-5, 1])
    %           ylim([10^-10, 1])
    %           set(gca, 'XScale', 'log', 'YScale', 'log')
    %           title(indPlotTitle)
    %--RcSw vs Se
    fig4 = figure(4);
    subplot(3,4,i)
    plot(sampleData(101:200, sampleData(201,:)==i), sampleData
        (1:100, sampleData(201,:)==i), 'Color',lineColor(i,:))
    xlabel('S_e')
    ylabel('r_c(S_w) / r_c(S_w=1)')
    indPlotTitle = sprintf('Rock Type %i', i);
```

```
        %set(gca, 'XScale', 'log', 'YScale', 'log')
        title(indPlotTitle)
        %--RcSw vs Se - Contact Angle
        %           fig4 = figure(4);
        %           subplot(3,4,i)
        %           plot(sampleData(101:200, sampleData(201,:)==i),
            sampleData(1:100, sampleData(201,:)==i), 'Color',
            lineColor(i,:))
        %           xlabel('S_e')
        %           ylabel('rcSw / max(rcSw)')
        %           indPlotTitle = sprintf('Rock Type %i', i);
        %           set(gca, 'XScale', 'log', 'YScale', 'log')
        %           title(indPlotTitle)
    end
end



%Uses curve clustering to automate the selection of our data set
    using the
%TEM-Function
if autoCluster == 2


    %--Must set path for CCT toolbox before it can be used.

    %Sets X data for curve cluster toolbox
```

```
    trajs.X = temData;
    %Sets Y data for curve cluster toolbox
    trajs.Y = swData;
    %Sets the clustering method
    ops.method = 'srm_d';
    %Sets dimensions of data, in this case two.
    ops.order = 4;
    %Sets the number of clusters
    ops.K = 8;
    %Sets normalization
    ops.zero = 'none';
    %Sets number of iterations
    %ops.NumEMStarts = 3;
    ops.TrainLhood = 50;
    %Clusters data
    model = curve_clust(trajs,ops);

    %Loads in manually selected curve clusters
    manClusters = model.C;
    %Appends cluster number to bottom of dataset
    sampleData_2(201,:) = manClusters';

    %--Indvidual Clustered Curves
    for i = 1:max(manClusters)
        %-- Sw vs Krw
        %           fig4 = figure(4);
        %           subplot(3,4,i)
```

```
%           plot(totSw(1:100, totSw(101,:)==i), totKrw(1:100, totKrw(101,:)==i), 'Color',lineColor(i,:))
%           xlabel('S_w')
%           ylabel('k_{rw}')
%           indPlotTitle = sprintf('Rock Type %i', i);
%           xlim([10^-5, 1])
%           ylim([10^-10, 1])
%           set(gca, 'XScale', 'log', 'YScale', 'log')
%           title(indPlotTitle)
%--TEM vs Sw
fig4 = figure(4);
subplot(3,3,i)
plot(sampleData_2(101:200, sampleData_2(201,:)==i), sampleData_2(1:100, sampleData_2(201,:)==i), 'Color', lineColor(i,:))
xlabel('S_w')
ylabel('TEM [D/cp]')
indPlotTitle = sprintf('Type %i', i);
%set(gca, 'YScale', 'log')
title(indPlotTitle)
%--RcSw vs Se - Contact Angle
%           fig4 = figure(4);
%           subplot(3,4,i)
%           plot(sampleData(101:200, sampleData(201,:)==i), sampleData(1:100, sampleData(201,:)==i), 'Color', lineColor(i,:))
%           xlabel('S_e')
```

```
        %           ylabel('rcSw / max(rcSw)')
        %           indPlotTitle = sprintf('Rock Type %i', i);
        %           set(gca, 'XScale', 'log', 'YScale', 'log')
        %           title(indPlotTitle)
    end
end

%% Total Data Plots

%Total single phase data plots
fig2 = figure(2);
%---Porosity vs Absolute Permeability
subplot(1,2,1)
hold on
plot(singlePhaseData(:,2), singlePhaseData(:,3), '.k', 'MarkerSize
    ',12)
xlabel('Porosity', 'FontSize', 16)
ylabel('Permeability [m^2]', 'FontSize', 16)
xlim([0.05, 0.35])
xticks([0.05, 0.15, 0.25, 0.35]);
xticklabels({'0.05', '0.15', '0.25', '0.35'})
xtickangle(30)
ylim([10e-16, 10e-10])
yticks([10e-16, 10e-14, 10e-12, 10e-10]);
yticklabels({'10e-16', '10e-14', '10e-12', '10e-10'})
set(gca, 'FontSize',14, 'YScale', 'log')
box on
```

```
%---Porosity vs Formation Factor
subplot(1,2,2)
plot(singlePhaseData(:,2), singlePhaseData(:,4), '.k', 'MarkerSize
    ',12)
xlabel('Porosity', 'FontSize', 16)
ylabel('Formation Factor', 'FontSize', 16)
xlim([0.05, 0.35])
xticks([0.05, 0.15, 0.25, 0.35]);
xticklabels({'0.05', '0.15', '0.25', '0.35'})
xtickangle(30)
ylim([5, 2*10^3])
yticks([5, 10^1, 10^2, 10^3, 2*10^3]);
yticklabels({'5', '10', '100', '1000', '2000'})
set(gca, 'FontSize',14, 'YScale', 'log')

%% Clustering Plots

close all

%Loads in manually selected curve clusters
manClusters = load('clusters.csv');
%Sets clusters to be the same as Ghanbarian, 2019 clusters for
    testing
%manClusters = ghanClusters(:,2);
%Appends cluster number to bottom of dataset
sampleData(201,:) = manClusters';
%---For loop to plotted clustered data
```

```
for i = 1:246

    fig3 = figure(3);
    %       subplot(1,2,1)
    hold on
    plot(totSw(:,i), totKrw(:,i), 'Color', lineColor(manClusters(i
        ),:))
    set(gca, 'FontSize', 14)
    xlabel('S_w', 'FontSize', 16)
    ylabel('K_{rw}', 'FontSize', 16)
    set(gca, 'YScale', 'log')
    %
    %       subplot(1,2,2)
    %       hold on
    %       plot(sampleData(101:200,i), sampleData(1:100,i), 'Color
        ', lineColor(manClusters(i),:))
    %       xlabel('S_e', 'FontSize', 16)
    %       ylabel('r_c(S_w) / r_c(S_w=1)', 'FontSize', 16)

    %--TEM-Function
    %       totKrw(:,i) = ((singlePhaseData(i,3)
        .*1013249965828.1448) .* totKrw(:,i)) ./ singlePhaseData(i
        ,2);

end

totSw(101,:) = manClusters';
```

```
totKrw(101,:) = manClusters';


%--Indvidual Clustered Curves
for i = 1:max(manClusters)
    %-- Sw vs Krw
    %     fig4 = figure(4);
    %     subplot(3,4,i)
    %     plot(totSw(1:100, totSw(101,:)==i), totKrw(1:100, totKrw
        (101,:)==i), 'Color',lineColor(i,:))
    %     xlabel('S_w')
    %     ylabel('k_{rw}')
    %     indPlotTitle = sprintf('Rock Type %i', i);
    %     xlim([10^-5, 1])
    %     ylim([10^-10, 1])
    %     set(gca, 'SScale', 'log', 'YScale', 'log')
    %     title(indPlotTitle)
    %--RcSw vs Se
    %     fig4 = figure(4);
    %     subplot(3,4,i)
    %     plot(sampleData(101:200, sampleData(201,:)==i),
        sampleData(1:100, sampleData(201,:)==i), 'Color',lineColor(
        i,:))
    %     xlabel('S_e')
    %     ylabel('r_c(S_w) / r_c(S_w=1)')
    %     indPlotTitle = sprintf('Rock Type %i', i);
    %    %set(gca, 'XScale', 'log', 'YScale', 'log')
```

```
    %       title(indPlotTitle)
    %--RcSw vs Se - Contact Angle
    %           fig4 = figure(4);
    %           subplot(3,4,i)
    %           plot(sampleData(101:200, sampleData(201,:)==i),
        sampleData(1:100, sampleData(201,:)==i), 'Color',lineColor(
        i,:))
    %           xlabel('S_e')
    %           ylabel('rcSw / max(rcSw)')
    %           indPlotTitle = sprintf('Rock Type %i', i);
    %           set(gca, 'XScale', 'log', 'YScale', 'log')
    %           title(indPlotTitle)
    %--TEM vs Sw
    %       fig4 = figure(4);
    %       subplot(3,4,i)
    %       plot(totSw(1:100, totSw(101,:)==i), totKrw(1:100, totKrw
        (101,:)==i), 'Color',lineColor(i,:))
    %       xlabel('S_w')
    %       ylabel('TEM [D/cp]')
    %       indPlotTitle = sprintf('Rock Type %i', i);
    %       title(indPlotTitle)
end

conAngle = zeros(1,246);
conAngle(121:240) = 60;
conAngleCounter_0 = zeros(1,12);
conAngleCounter_60 = zeros(1,12);
```

```
%---Individual Clustered Curves - Contact Angle
for i = 1:246
    fig4 = figure(4);
    subplot(3,4,manClusters(i))
    if conAngle(i) == 0
        plot(sampleData(101:200,i), sampleData(1:100,i), 'k')
        hold on
        xlabel('S_e')
        ylabel('r_c(S_w) / r_c(S_w=1)')
        indPlotTitle = sprintf('Rock Type %i', manClusters(i));
        %set(gca, 'XScale', 'log', 'YScale', 'log')
        title(indPlotTitle)
        conAngleCounter_0(manClusters(i)) = conAngleCounter_0(
            manClusters(i)) + 1;
    else
        plot(sampleData(101:200,i), sampleData(1:100,i), 'r')
        hold on
        xlabel('S_e')
        ylabel('r_c(S_w) / r_c(S_w=1)')
        indPlotTitle = sprintf('Rock Type %i', manClusters(i));
        %set(gca, 'XScale', 'log', 'YScale', 'log')
        title(indPlotTitle)
        conAngleCounter_60(manClusters(i)) = conAngleCounter_60(
            manClusters(i)) + 1;
    end
end
```

```
%---Single Phase Clustering Plots
for i = 1:246
    fig5 = figure(5);
    %---Porosity vs Absolute Permeability
    %     subplot(1,2,1)
    %     hold on
    %     plot(singlePhaseData(i,2), singlePhaseData(i,3), '.', '
        MarkerSize',12, 'Color', lineColor(manClusters(i),:))
    %     xlabel('Porosity', 'FontSize', 16)
    %     ylabel('Absolute Permeability [m^2]', 'FontSize', 16)
    %     xlim([0.05 0.35])
    %     set(gca, 'FontSize',14, 'YScale', 'log')
    %---Porosity vs Formation Factor
    %     subplot(1,2,2)
    %     hold on
    %     plot(singlePhaseData(i,2), singlePhaseData(i,4), '.', '
        MarkerSize',12,'Color', lineColor(manClusters(i),:))
    %     xlabel('Porosity', 'FontSize', 16)
    %     ylabel('Formation Factor', 'FontSize', 16)
    %     xlim([0.05 0.35])
    %     set(gca, 'FontSize',14, 'YScale', 'log')
    %---1/F vs Absolute Permeability
    subplot(1,2,1)
    hold on
    plot(1/singlePhaseData(i,4), singlePhaseData(i,3), '.k', '
        MarkerSize', 12)
```

```
    xlabel('1/F', 'FontSize', 16)
    ylabel('Permeability [m^2]', 'FontSize', 16)
    set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')
    subplot(1,2,2)
    hold on
    plot(1/singlePhaseData(i,4), singlePhaseData(i,3), '.', '
        MarkerSize',12,'Color', lineColor(manClusters(i),:))
    xlabel('1/F', 'FontSize', 16)
    ylabel('Permeability [m^2]', 'FontSize', 16)
    set(gca, 'FontSize',14, 'XScale', 'log', 'YScale', 'log')
end

%% Representative Curves

%--Plots Manually picked representative curves
for i = 1:max(manClusters)

    %--Averaging code
    %Creates prerepresentative rcSw vs Se Curves
    repCurve = [mean(sampleData(1:100, sampleData(201,:)==i),2),
        mean(sampleData(101:200, sampleData(201,:)==i),2)];

    %--Plots data
    fig6 = figure(6);
    sgtitle('Representative Rock Types', 'FontSize', 24)
    %Representative Curves
    plot(repCurve(:,1), repCurve(:,2), 'Color',lineColor(i,:), '
```

```
        LineWidth',1.25)
    hold on
    xlabel('S_e', 'FontSize', 16)
    ylabel('r_c(S_w)/max(r_c(S_w))', 'FontSize', 16)
    set(gca, 'FontSize',14)
    legend({'Rock Type 1', 'Rock Type 2', 'Rock Type 3', 'Rock
        Type 4', 'Rock Type 5', 'Rock Type 6', 'Rock Type 7', 'Rock
         Type 8', 'Rock Type 9', 'Rock Type 10', 'Rock Type 11', '
        Rock Type 12'}, 'FontSize',16, 'location', 'eastoutside');

end


%% Legend Plotting

for i = 1:12
    fig7 = figure(7);
    hold on
    plot(i, i, '.', 'MarkerSize',12,'Color', lineColor(i,:))
    legend({'Rock Type 1', 'Rock Type 2', 'Rock Type 3', 'Rock
        Type 4', 'Rock Type 5', 'Rock Type 6', 'Rock Type 7', 'Rock
         Type 8', 'Rock Type 9', 'Rock Type 10', 'Rock Type 11', '
        Rock Type 12'}, 'FontSize',16, 'location', 'eastoutside');

    fig8 = figure(8);
    hold on
    plot(rand(1,10), rand(1,10), 'Color', lineColor(i,:))
```

```
    legend({'Rock Type 1', 'Rock Type 2', 'Rock Type 3', 'Rock Type 4', 'Rock Type 5', 'Rock Type 6', 'Rock Type 7', 'Rock Type 8', 'Rock Type 9', 'Rock Type 10', 'Rock Type 11', 'Rock Type 12'}, 'FontSize',16, 'location', 'eastoutside');
end


%% Archies Law

%loads in archie's law clusters, and discards the two samples that are
%outliers. Note however, both outliers are real-world samples.
archieClusters = load('archieClusters.csv');
porosity = singlePhaseData(:,2);
% porosity(241) = [];
% porosity(245) = [];
formFactor = singlePhaseData(:,4);
% formFactor(241) = 9;
% formFactor(245) = 9;
% archieClusters(archieClusters == 9) = [];
% indM_Values = nan(8,2);

%Calculates a and m values for archie's law
for i = 1:8
    %Traditional Archie's Law
    xData = porosity(archieClusters == i);
    yData = formFactor(archieClusters == i);
```

```
    [f, gof] = fit(xData, yData, 'a*x^m');
    m = -log(1./yData) ./ log(xData);
    indM_Values(i,:) = [min(m), max(m)];
    aValues(i) = f.a;
    mValues(i) = f.m;
    %r2Temp = corrcoef(xData, yData);
    rSquared(i) = gof.rsquare;
    rmseValues(i) = gof.rmse;

end

aValues = aValues';
mValues = mValues'*-1;
rSquared = rSquared';
rmseValues = rmseValues';

m = -log(formFactor) ./ log(porosity);

%---Muller-Huber2015 Method
poreRatio = singlePhaseData(:,2) .* singlePhaseData(:,4);
mullerValues = 1 - log10(poreRatio) ./ log10(singlePhaseData(:,2));


%Plotting colors for archies law
archieColor = ['#E20000'; '#9B1010'; '#f58231'; '#ffe119'; '#99D649'; ...
```

```
    '#1D9128'; '#013220'; '#42d4f4'; '#FF1AAD'];

%---Archies Law Clusters
fig8 = figure(8);
subplot(1,2,1)
hold on
plot(singlePhaseData(:,2), singlePhaseData(:,3), '.k', 'MarkerSize
    ',12)
xlabel('Porosity', 'FontSize', 16)
ylabel('Permeability [m^2]', 'FontSize', 16)
xlim([0.05, 0.35])
xticks([0.05, 0.15, 0.25, 0.35]);
xticklabels({'0.05', '0.15', '0.25', '0.35'})
xtickangle(30)
ylim([10e-16, 10e-10])
yticks([10e-16, 10e-14, 10e-12, 10e-10]);
yticklabels({'10e-16', '10e-14', '10e-12', '10e-10'})
set(gca, 'FontSize',14, 'YScale', 'log')
box on
subplot(1,2,2)
hold on
gscatter(porosity, formFactor, archieClusters, archieColor)
xlabel('Porosity', 'FontSize', 16)
ylabel('Formation Factor', 'FontSize', 16)
xlim([0.05, 0.35])
xticks([0.05, 0.15, 0.25, 0.35]);
xticklabels({'0.05', '0.15', '0.25', '0.35'})
```

```
xtickangle(30)
ylim([5, 2*10^3])
yticks([5, 10^1, 10^2, 10^3, 2*10^3]);
yticklabels({'5', '10', '100', '1000', '2000'})
%legend({'1','2','3','4','5','6','7','8',},'Location', '
    eastoutside')
legend off
box on
set(gca, 'FontSize',14, 'YScale', 'log')



%% Nelson Plot

%Plotting Area's

%Consolidated Clays
group1 = [0.2, 1.00E-05;
     0.55, 1.00E-05;
     0.55, 1.00E-07;
     0.2, 1.00E-07;
     0.2, 1.00E-05];

%Consolidated Sands & Carbonates
group2 = [0, 1.00E-04;
     0.35, 1.00E-04;
     0.35, 10;
```

```
    0, 10;
    0, 1.00E-04];

%Unconsolidated Sands
group3 = [0.45, 0.1;
    0.45, 1000;
    0.1, 1000;
    0.1, 0.1;
    0.45, 0.1];


%---Ploting Code
fig1 = figure(1);

%Sets color for gragh
colororder({'k','k'})
%Sets Left Axis
yyaxis left
hold on
plot(group1(:,1), group1(:,2), '-r', 'LineWidth', 2)
plot(group2(:,1), group2(:,2), '-g', 'LineWidth', 2)
plot(group3(:,1), group3(:,2), '-b', 'LineWidth', 2)
plot(singlePhaseData(:,2), singlePhaseData(:,3).*1013250000000, '.
    k', 'MarkerSize',12')
xlim([0,0.7])
ylim([1e-7, 1000])
xticks([0,0.1,0.2,0.3,0.4,0.5,0.6,0.7])
```

```
xticklabels({'0','0.1','0.2','0.3','0.4','0.5','0.6','0.7'})
yticks([10e-8, 10e-6, 10e-4, 10e-2, 10e0, 10e2])
yticklabels({'10e-8', '10e-6', '10e-4', '10e-2', '10e0', '10e2'})
xlabel('Porosity', 'FontSize',16)
ylabel('Permeability [D]', 'FontSize',16)
set(gca, 'YScale', 'log')
%Sets Right Axis
yyaxis right
hold on
%plot(singlePhaseData(:,2), singlePhaseData(:,3), '.r', '
    MarkerSize',12')
xlim([0,0.7])
ylim([10e-20, 10e-10])
yticks([10e-20, 10e-18, 10e-16, 10e-14, 10e-12, 10e-10])
yticklabels({'10e-20','10e-18', '10e-16', '10e-14', '10e-12', '10e
    -10'})
xlabel('Porosity', 'FontSize',16)
ylabel('Permeability [m^2]', 'FontSize',16)
set(gca, 'YScale', 'log')

%% Ghanbarian, 2019 Equation

close all

lC = 2.825;
lC_Y = linspace(0.0001,1,100000)-0.001;
lC_X = lC^2 ./ (8 .* linspace(0.0001,1,100000)-0.001);
```

```
permData = singlePhaseData(:,3);%.*1013249965828.1448;
formData = 1./singlePhaseData(:,4);

[ide, C] = kmeans([formData, permData], 12);
ghanClusters = load('ghanbarianClusters.csv');
lC = nan(max(ghanClusters(:,2)),1);


%Finds characteristic length scale
for i = 1:max(ghanClusters(:,2))
    xData = formData(ghanClusters(:,2) == i);
    yData = permData(ghanClusters(:,2) == i);

    ft = fittype( 'poly1' );
    opts = fitoptions( 'Method', 'LinearLeastSquares' );
    opts.Lower = [-Inf -Inf];
    f = fit( xData, yData, ft, opts );

    lC(i) = sqrt(f.p1*8);

    xFit = linspace(min(1./singlePhaseData(:,4)), max(1./
        singlePhaseData(:,4)),10000);
    yFit = f.p1 .* xFit + f.p2;

    %     fig2 = figure(2);
    %     hold on
```

```
    %       plot(xData, yData, '.', 'MarkerSize', 12, 'Color', lineColor(i,:))
    %       %     plot(xFit, yFit)
    %       plot(f)
    %       xlabel('1/F', 'FontSize', 16)
    %       ylabel('Absolute Permeability [m^2]', 'FontSize', 16)
    %       %       xlim([7e-4, 10e0])
    %       %       ylim([2e-4, 1.4e2])
    %       set(gca, 'FontSize',14, 'YScale', 'log', 'XScale', 'log')
end

manClusters = load('clusters.csv');

close all

fig2 = figure(2);
hold on
plot(formData, permData, '.k', 'MarkerSize', 12)
xlabel('1/F', 'FontSize', 16)
ylabel('Permeability [m^2]', 'FontSize', 16)
xlim([10e-5,1])
ylim([10e-16,10e-10])
xticks([10e-5, 10e-4, 10e-3, 10e-2, 10e-1, 10e0]);
xticklabels({'10e-5', '10e-4', '10e-3', '10e-2', '10e-1', '10e0'})
xtickangle(60)
yticks([10e-16, 10e-14, 10e-12, 10e-10]);
```

```
yticklabels({'10e-16', '10e-14', '10e-12', '10e-10'})
set(gca, 'FontSize',14, 'YScale', 'log', 'XScale', 'log')
%print(fig2, 'ghanbarianPlot_Base', '-dpng', '-r300')

clf
fig2 = figure(2);
hold on
plotVar = gscatter(formData, permData, ghanClusters(:,2),
    lineColor);
xlabel('1/F', 'FontSize', 16)
ylabel('Permeability [m^2]', 'FontSize', 16)
xlim([10e-5,1])
ylim([10e-16,10e-10])
xticks([10e-5, 10e-4, 10e-3, 10e-2, 10e-1, 10e0]);
xticklabels({'10e-5', '10e-4', '10e-3', '10e-2', '10e-1', '10e0'})
xtickangle(60)
yticks([10e-16, 10e-14, 10e-12, 10e-10]);
yticklabels({'10e-16', '10e-14', '10e-12', '10e-10'})
set(gca, 'FontSize',14, 'YScale', 'log', 'XScale', 'log')
legend('Location', 'eastoutside')
%print(fig2, 'ghanbarianPlot_GhanClusters', '-dpng', '-r300')


clf
fig2 = figure(2);
hold on
gscatter(formData, permData, manClusters, lineColor)
```

```
xlabel('1/F', 'FontSize', 16)
ylabel('Permeability [m^2]', 'FontSize', 16)
xlim([10e-5,1])
ylim([10e-16,10e-10])
xticks([10e-5, 10e-4, 10e-3, 10e-2, 10e-1, 10e0]);
xticklabels({'10e-5', '10e-4', '10e-3', '10e-2', '10e-1', '10e0'})
xtickangle(60)
yticks([10e-16, 10e-14, 10e-12, 10e-10]);
yticklabels({'10e-16', '10e-14', '10e-12', '10e-10'})
set(gca, 'FontSize',14, 'YScale', 'log', 'XScale', 'log')
legend('Location', 'eastoutside')
%print(fig2, 'ghanbarianPlot_ManClusters', '-dpng', '-r300')


%---Plots
fig2 = figure(2);
hold on
%plot(1./singlePhaseData(:,4), singlePhaseData(:,3)
    .*1013249965828.1448, '.k', 'MarkerSize',12)
plot(linspace(0.0001,1,100000)-0.001, linspace(0.0001,1,100000)
    -0.001, 'b')
plot(lC_X, lC_Y, 'r')
xlim([min(1./singlePhaseData(:,4)), max(1./singlePhaseData(:,4))])
ylim([min(singlePhaseData(:,3).*1013249965828.1448), max(
    singlePhaseData(:,3).*1013249965828.1448)])
xlabel('1/F', 'FontSize', 16)
ylabel('Permeability [mD]', 'FontSize', 16)
```

```
set(gca, 'FontSize',14, 'YScale', 'log', 'XScale', 'log')

%% RQI & FZI Equations

%Calculates RZI values so that I can calculate FZI
rqi = 0.0314 * ((1013249965828.1448.*singlePhaseData(:,3)) ./
    singlePhaseData(:,2)).^(1/2);
%Calculates the void ratio of the samples phi/(1-phi)
voidRatio = singlePhaseData(:,2) ./ (1-singlePhaseData(:,2));
%Calculates FZI values
fzi = rqi .* voidRatio;
% Steps to find FZI groups outlined in Riazi, 2018
% avgVals = nan(246,2);
% avgVals(:,1) = (1013249965828.1448.*singlePhaseData(:,3)) .* swi
    ;
% avgVals(:,2) = 1:246';
% Sorted for min to max
% avgVals = sortrows(avgVals);
% [Y, E] = discretize(avgVals(:,2), 20);

Y2 = discretize(fzi, 12);



close all
fig1 = figure(1);
%gscatter(voidRatio, rqi, Y2)
```

```
plot((0.1./sqrt(singlePhaseData(:,2))),
    0.0314.*1013249965828.1448.*singlePhaseData(:,3), '.k', '
    MarkerSize', 12)
xlabel('Porosity', 'FontSize', 16)
ylabel('Absolute Permeability [m^2]', 'FontSize', 16)
xlim([0.01, 1])
ylim([0.01, 1])
%legend({'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster4', '
    Cluster 5'}, 'Location', 'best')
set(gca, 'FontSize',14, 'YScale', 'log', 'XScale', 'log')

%% Nishiyama, 2017 Method, Critical pore size vs permeability

close all

fig1 = figure(1);
hold on
plot(singlePhaseData(:,5).^2 .* singlePhaseData(:,2),
    singlePhaseData(:,3), '.k', 'MarkerSize', 12)
xlabel('r_c^2 * phi [m^2]', 'FontSize', 16)
ylabel('Permeability [m^2]', 'FontSize', 16)
set(gca, 'XScale', 'log', 'YScale', 'log', 'FontSize', 14)
% clf
% fig1 = figure(1);
% plot(singlePhaseData(:,5), singlePhaseData(:,2), '.k', '
    MarkerSize', 12)
% xlabel('Critical Pore Size [m]', 'FontSize', 16)
```

```
% ylabel('Porosity', 'FontSize', 16)
% set(gca, 'XScale', 'log', 'YScale', 'log', 'FontSize', 14)
%
% clf
% fig1 = figure(1);
% plot(singlePhaseData(:,5), singlePhaseData(:,4), '.k', '
    MarkerSize', 12)
% xlabel('Critical Pore Size [m]', 'FontSize', 16)
% ylabel('Formation Factor', 'FontSize', 16)
% set(gca, 'XScale', 'log', 'YScale', 'log', 'FontSize', 14)

%---[49,52,55,58,169,172,175,178]
%Networks that give weird values from rcSW^2/F

connNumber = load('connNumber.csv');

xData_2 = singlePhaseData(connNumber == 2, :);
yData_2 = xData_2(:,3);
xData_2 = xData_2(:,5) .^ 2 .* xData_2(:,2);

xData_4 = singlePhaseData(connNumber == 4, :);
yData_4 = xData_4(:,3);
xData_4 = xData_4(:,5) .^ 2 .* xData_4(:,2);

xData_6 = singlePhaseData(connNumber == 6, :);
yData_6 = xData_6(:,3);
xData_6 = xData_6(:,5) .^ 2 .* xData_6(:,2);
```

```
%Collapsed data using r_c^2/F
xData = singlePhaseData(1:240,5).^2./singlePhaseData(1:240,4);
yData = singlePhaseData(1:240,3);
[f, gof] = fit(xData, yData, 'power1');
yFit = xData .* (xData\yData);

close all

%---r_c VS Perm
fig1 = figure(1);
hold on
gscatter(singlePhaseData(1:240,5), singlePhaseData(1:240,3),
    connNumber(1:240))
xlabel('r_c(S_w = 1) [m]', 'FontSize', 16)
ylabel('Permeability [m^2]', 'FontSize', 16)
xlim([10^-6, 10^-4])
ylim([10^-16, 10^-9])
xticks([10^-6, 10^-5, 10^-4]);
xticklabels({'10e-7', '10e-6', '10e-5'})
xtickangle(30)
yticks([10^-16, 10^-14, 10^-12, 10^-10, 10^-9]);
yticklabels({'10e-17', '10e-15', '10e-13', '10e-11', '10e-10'})
set(gca, 'XScale', 'log', 'YScale', 'log', 'FontSize', 14)
legend({'Z-Value: 2', 'Z-Value: 4', 'Z-Value: 6'}, 'Location', '
    best');
%print(fig1, 'critPore_VS_Perm_1', '-dpng', '-r300')
```

```
%---r_c^2/F VS Perm
clf
fig1 = figure(1);
hold on
gscatter(singlePhaseData(1:240,5).^2./singlePhaseData(1:240,4),
    singlePhaseData(1:240,3), connNumber(1:240))
xlabel('r_c(S_w = 1)^2/F [m^2]', 'FontSize', 16)
%lineFit = plot(xData, yFit, '-k', 'LineWidth', 2);
ylabel('Permeability [m^2]', 'FontSize', 16)
%legend(lineFit, '0.1614(r_c^2/F)', 'Location', 'best', 'FontSize
    ',16)
xlim([10^-15, 10^-9])
ylim([10^-16, 10^-9])
xticks([10^-15, 10^-14, 10^-13, 10^-12, 10^-11, 10^-10, 10^-9]);
xtickangle(30)
xticklabels({'10e-16', '10e-15', '10e-14', '10e-13', '10e-12', '10
    e-11', '10e-10'})
yticks([10^-16, 10^-14, 10^-12, 10^-10, 10^-9]);
yticklabels({'10e-17', '10e-15', '10e-13', '10e-11', '10e-10'})
set(gca, 'XScale', 'log', 'YScale', 'log', 'FontSize', 14)
legend off
%print(fig1, 'critPore_VS_Perm_2', '-dpng', '-r300')

%---r_c vs Phi
clf
fig1 = figure(1);
```

```
hold on
gscatter(singlePhaseData(1:240,5), singlePhaseData(1:240,4),
    connNumber(1:240))
xlabel('r_c(S_w = 1) [m]', 'FontSize', 16)
ylabel('Formation Factor', 'FontSize', 16)
xlim([10^-6, 10^-4])
ylim([10^0, 2*10^3])
xticks([10^-6, 10^-5, 10^-4]);
xticklabels({'10e-7', '10e-6', '10e-5'})
xtickangle(30)
yticks([10^0, 10^1, 10^2, 10^3, 2*10^3]);
yticklabels({'1', '10', '100', '1000', '2000'})
set(gca, 'XScale', 'log', 'YScale', 'log', 'FontSize', 14)
legend off
%print(fig1, 'critPore_VS_Form', '-dpng', '-r300')

%---r_c VS Form Factor
clf
fig1 = figure(1);
hold on
gscatter(singlePhaseData(1:240,5), singlePhaseData(1:240,2),
    connNumber(1:240))
xlabel('r_c(S_w = 1) [m]', 'FontSize', 16)
ylabel('Porosity', 'FontSize', 16)
xlim([10^-6, 10^-4])
ylim([0.05, 0.35])
xticks([10^-6, 10^-5, 10^-4]);
```

```
xticklabels({'10e-7', '10e-6', '10e-5'})
xtickangle(30)
yticks([0.05, 0.15, 0.25, 0.35]);
yticklabels({'0.05', '0.15', '0.25', '0.35'})
set(gca, 'XScale', 'log', 'YScale', 'log', 'FontSize', 14)
legend off
%print(fig1, 'critPore_VS_Phi', '-dpng', '-r300')

toc
```