Image-based Modeling of Flow through Porous Media: Development of Multiscale Techniques for the Pore Level

Timothy Wayne Thibodeaux
Louisiana State University and Agricultural and Mechanical College, timothythibodeaux@gmail.com

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_dissertations

Part of the Computational Engineering Commons, and the Petroleum Engineering Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_dissertations/4197

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
IMAGE-BASED MODELING OF FLOW THROUGH POROUS MEDIA:
DEVELOPMENT OF MULTISCALE TECHNIQUES FOR THE PORE LEVEL

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

Chemical Engineering

by
Timothy Wayne Thibodeaux
B.S., University of Louisiana at Lafayette, 2010
May 2018
ACKNOWLEDGMENTS

The research described in this work was funded by BP and Schlumberger as part of the PoreSim Research Consortium. The microCT data of the reservoir sandstone used in Chapter 2 was provided by Dr. Joanne Fredrich at BP. The Dolomite sample was imaged at the National Synchrotron Light Source (NSLS) X2B beamline at Brookhaven National Laboratory as part of a project funded by ExxonMobil Upstream Research Company. We thank Prof. Clinton Willson for help obtaining this data set.

Many thanks to Dr. Karsten Thompson for the years of support, advice, and patience, and to my fellow researchers, including Nathan, Qiang, Amin, Saade, Yijie, Paula, Godfrey, Layne, Dongxing, Jack, and Chris. Very special thanks to my family for their complete, unceasing support and encouragement throughout my very long education: to Maddy, for her continual love and friendship, support of my academic pursuits, the many sacrifices she has endured in the process, and for balancing my life and teaching me to prioritize what really matters; to Mom for always being an example of unconditional love and self-sacrifice; to Pops for teaching me to work hard and ethically, and often putting my anxieties into perspective; to Sarah for always setting the bar high and inspiring me to reach for it; and to my Olivia Jean for making the last year fun and joyous, despite the stressful process of finishing this work.
TABLE OF CONTENTS

ACKNOWLEDGMENTS ........................................................................................................ii

ABSTRACT .......................................................................................................................... iv

1 INTRODUCTION .................................................................................................................. 1
  1.1 Objectives ....................................................................................................................... 1
  1.2 Chapter Descriptions ................................................................................................. 2

2 RAPID ESTIMATION OF ESSENTIAL POROUS MEDIA PROPERTIES USING IMAGE-BASED PORE-SCALE NETWORK MODELING ................................................................. 3
  2.1 Introduction ................................................................................................................... 3
  2.2 Background .................................................................................................................. 4
  2.3 Methodology ............................................................................................................... 8
  2.4 Results and Discussion ............................................................................................. 17
  2.5 Conclusions .............................................................................................................. 26

3 NETWORK-FEM MULTISCALE MODELING ...................................................................... 28
  3.1 Introduction ................................................................................................................... 28
  3.2 Background .................................................................................................................. 28
  3.3 Methodology ............................................................................................................... 67
  3.4 Results and Discussion ............................................................................................. 82
  3.5 Conclusions .............................................................................................................. 101

4 MULTISCALE MODELING OF PARTICLE FILTRATION .................................................... 106
  4.1 Introduction ................................................................................................................... 106
  4.2 Background .................................................................................................................. 106
  4.3 Methodology ............................................................................................................... 110
  4.4 Results and Discussion ............................................................................................. 118
  4.5 Conclusions .............................................................................................................. 129

5 SUMMARY ....................................................................................................................... 131
  5.1 Recommendations and Future Work .................................................................... 131

REFERENCES ..................................................................................................................... 138

APPENDIX A. ACS RIGHTS LINK REUSE REQUEST AND PERMISSION ...................................... 151

APPENDIX B. MASS BALANCE IN FEM SOLUTION ................................................................ 153

APPENDIX C. ADDITIONAL DATA ....................................................................................... 158

APPENDIX D. HYDRAULIC CONDUCTIVITY APPROXIMATION FORMULAS .............................. 160

VITA .................................................................................................................................... 171
ABSTRACT

Increasingly, imaging technology allows porous media problems to be modeled at microscopic and sub-microscopic levels with finer resolution. However, the physical domain size required to be representative of the media prohibits comprehensive micro-scale simulation. A hybrid or multiscale approach is necessary to overcome this challenge. In this work, a technique was developed for determining the characteristic scales of porous materials, and a multiscale modeling methodology was developed to better understand the interaction/dependence of phenomena occurring at different microscopic scales. The multiscale method couples microscopic simulations at the pore and sub-pore scales. Network modeling is a common pore-scale technique which employs severe assumptions, making it more computationally efficient than direct numerical simulation, enabling simulation over larger length scales. However, microscopic features of the medium are lost in the discretization of a material into a network of interconnected pores and throats. In contrast, detailed microstructure and flow patterns can be captured by modern meshing and direct numerical simulation techniques, but these models are computationally expensive. In this study, a data-driven multiscale technique has been developed that couples the two types of models, taking advantage of the benefits of each. Specifically, an image-based physically-representative pore network model is coupled to an FEM (finite element method) solver that operates on unstructured meshes capable of resolving details orders of magnitude smaller than the pore size. In addition to allowing simulation at multiple scales, the current implementation couples the models using a machine learning approach, where results from the FEM model are used to learn network model parameters. Examples of the model operating on real materials are given that demonstrate improvements in network modeling enabled by the multiscale framework. The framework enables more advanced multiscale and multiphysics modeling – an application to particle straining problems is shown. More realistic network filtration simulations are possible by incorporating information from the sub-pore-scale. New insights into the size exclusion mechanism of particulate filtration were gained in the process of generating data for machine learning of conductivity reduction due to particle trapping. Additional tests are required to validate the multiscale network filtration model, and compare with experimental findings in literature.
1 INTRODUCTION

The study of flow in porous media has applications in groundwater contaminant transport, petroleum reservoir analysis, geothermal energy production, and several more disciplines which are important to our society. Analyzing these large-scale systems (e.g., reservoirs and aquifers) require petrophysical properties of the rock formations, which are dictated by smaller-scale phenomena. These properties include porosity, permeability (absolute, tensor, and relative), formation factor, and dispersion coefficient, to name a few. They are typically determined by core flood experiments which are extremely time-consuming (some core analyses take months or even over a year). Image-based, predictive modeling of these materials is becoming a popular way to reduce this burden, with some properties requiring between minutes and days for calculation. Another advantage over experimental measurements is that a single core can be used for computing multiple properties simultaneously (on multiple CPUs) whereas some experiments cause physical damage which permits only a single experiment to be run on each sample.

In this work, techniques are developed for image-based simulation of fluid flow in porous media. These techniques allow computation of petrophysical properties. Specifically, the focus is on pore-scale modeling and studying the phenomena occurring in porous materials at various length scales. We hypothesize the following statements: Network modeling is an ideal tool for rapid assessment of the characteristic length of porous media, and can therefore be used to determine whether an image is of adequate size for continuum-scale property prediction. Coupling pore-scale and sub-pore-scale models will provide better insight into multiscale transport phenomena in porous media than modeling solely at either scale. The relatively large maximum domain size of pore-scale models and robust accuracy and conformance to pore geometries of streamline-scale models are both important to predicting macroscopic properties of porous materials. Given enough data (training examples consisting of geometric and network-derived features and FEM-computed hydraulic conductivities for a variety of throat shapes/sizes/etc.), an appropriate machine learning algorithm would be able to derive an expression which can accurately predict hydraulic conductance values for network throats. The machine learning framework for coupling these models can then be extended to other phenomena which can be accurately simulated by FEM, such as particle straining and adsorption/desorption, reactive flow, non-isothermal flow, and multiphase flow.

1.1 Objectives

1. Build upon existing in-house pore-scale modeling codes to add functionality and increase generality in order to enable coupling, successive calling, and efficient data handling.

2. Create a fast algorithm to (1) determine the characteristic length scale for continuum parameters, and (2) assess the computational domain size required for image-based modeling of porous materials.
3. Develop a data-driven framework for pore-level multiscale modeling to incorporate sub-pore scale physics into pore scale models, specifically for image-based models of real porous materials.

4. Use the framework for learning prediction expressions for the network model parameter, absolute hydraulic conductivity. This will provide insight into the significance of various pore-space properties on resistance to flow, and will show the variation of significance between different porous materials.

5. Apply the framework to multiscale particle filtration problems

1.2 Chapter Descriptions

Chapter 2 showcases two image-based pore network modeling algorithms which efficiently compute macroscopic properties of porous materials – characteristic length and permeability tensor, specifically. This work was published in the American Chemistry Society journal, Industrial and Engineering Chemistry Research (Thibodeaux, Sheng et al. 2015) and is reproduced here with permission (see APPENDIX A). Being a more industrial application-based work, the background on the network algorithm used is brief, but a more detailed discussion is included in 3. Chapter 3 presents the multiscale coupling of the FEM and Pore Network models at the pore-scale. A machine learning approach was used for predicting network parameters from FEM Stokes flow solutions. Chapter 4 covers an application of the multiscale methodology presented in Chapter 3 to particle filtration: a network filtration model was adapted to utilize the learned parameter prediction in order to remove one of the inherent oversimplified assumptions, and the multiscale framework was used to generate data for conductivity reduction due to particle straining. Chapter 5 provides some conclusions on the material presented in the previous chapters as well as recommendations and proposals for future work on the topics discussed.
Physically representative network models have been used for years for investigating pore-scale behavior in porous materials. However, the technology has remained largely in the research domain, with limited application to industrial processes. In this work, we introduce two algorithms that are derived from well-tested network modeling techniques, but provide unique capabilities for rapid assessment of important porous material properties from high-resolution 3D images.

The first is a fast, fully automated algorithm for determining the characteristic length scale for three continuum parameters: porosity, permeability, and electrical resistivity (or formation factor). This information is important for assessing the size of the computational domain needed for image-based modeling. It operates by computing relevant continuum properties from consecutively smaller sub-domains extracted from a larger pore network model. Resulting data trends are plotted so that the relevant characteristic lengths can be inferred. The second algorithm is a novel approach for computing the full permeability tensor, which is important for quantifying whether a material is isotropic. An artificial anisotropic porous medium was generated with a known principal flow direction not aligned with the Cartesian coordinate axes. The algorithm was validated by its ability to determine both the permeability tensor and the principal flow direction accurately and efficiently. Both network modeling algorithms were applied to networks derived from microCT images of real materials.

2.1 Introduction

Porous media are found in a wide variety of industrial processes including oil and gas recovery, catalysis, chemical separations, and advanced materials manufacturing. In most of these applications the pore structure is microscopic (nanometer or micrometer scale) and is generally orders of magnitude smaller than the characteristic scales governing the overall processes or bulk materials. Hence, for practical reasons porous media are commonly modeled as continuum materials, in which case they must be assigned attributes that reflect their porous structure: porosity, permeability, pore-size distribution, specific surface area, etc. While this separation of scales is necessary, it often creates a disconnect between the larger-scale description (used for practical modeling) and the pore-scale description (necessary for modeling the most fundamental physics).

Pore-scale modeling operates at length scales small enough to independently resolve the pore space and solid phase. Historically this approach was used with idealized structures in order to help understand phenomenological behavior. However, starting in the late 1990’s, high-resolution, three-dimensional imaging techniques such as x-ray microtomography became widely available. This transformed pore-scale modeling by allowing digital images of real materials to be used as the basis for computational modeling. This advance was critical because it meant that the heterogeneity and spatial correlations found in real materials (and which ultimately impact macroscopic properties) could be accounted for implicitly, and is allowing image-based modeling to develop into a powerful analytic tool.

Modern pore-scale, image-based modeling encompasses a very wide range of techniques and applications, from well-tested algorithms for predictive modeling to advanced research codes that are used to study complex multiphase behaviors. Unfortunately, even the mature techniques have not found their way to industrial applications as quickly as one would hope. The problem is largely that this type of analysis remains fairly specialized: modeling codes are typically generated by independent research groups or specialized consulting companies, and the workflow from 3D imaging to quantitative materials characterization requires specialized expertise.

This paper introduces two new algorithms that are ideal for applied characterization of porous materials in an image-based modeling workflow. The underlying properties are based on single-phase behavior; hence, the foundation for the algorithms is well understood and tested. However, new implementations and mathematics are described below, with an emphasis on applied modeling. The first of the two algorithms provides rapid, fully automated assessment of the critical characteristic-scale for three important continuum-phase parameters: porosity, permeability, and electrical resistivity. The second allows rapid computation of the full permeability tensor, which is an essential property of materials with anisotropic pore structure, especially in cases where the orientation of the material during imaging may not have been aligned with the principal directions of permeability.

In the remainder of the paper, we provide background on the specific parameters quantified by this approach, describe the porous materials used in the analysis, present the algorithms in detail, and present examples using both computer-generated and real porous media.

2.2 Background

2.2.1 Permeability

The Reynolds number (Re) is the ratio of inertial to viscous forces, and is used to quantify the flow regimes occurring in a medium.

For porous media, it is defined as
where \( \rho \) is the fluid density, \( V \) is the superficial velocity – the flow rate (\( Q \)) divided by cross-sectional area (\( A \)), \( \mu \) is the fluid viscosity, and \( d \) is the characteristic length of the material, which has often been equated to the diameter of the grains which comprise it. It is widely accepted that Darcy’s Law is valid for quantifying low Reynolds number flows through porous media (i.e. when viscous forces dominate and inertial forces are negligible), which are common considering how tiny the characteristic length scale is and how large the pressure gradient would have to be to cause high velocities within porous media. Darcy’s law dictates that the flow rate through a porous medium increases linearly with applied pressure gradient (\( dP/dx \)) via a proportionality coefficient, permeability (\( K \)), which can be considered an intrinsic parameter of the medium at a given state:

\[
V = \frac{Q}{A} = -\frac{K}{\mu} \left( \frac{dP}{dx} \right)
\]

Therefore, permeability is a measure of a medium’s ability to transmit fluids. Traditionally, the permeabilities of geologic samples have been found via core-flood experiments, where a cylindrical core of the material is subjected to a constant pressure gradient and volumetric flow is measured over time. These traditional methods are limited by a number of factors: they are time consuming, usually a limited number of tests can be performed on any one sample, the details of the pore-scale flow distribution are rarely known, and measurement of anisotropy is difficult (Dullien 1991). Digital rock technology allows the prediction of various continuum properties without such experiments. Permeability can be found by numerical simulations analogous to the experimental procedure (i.e. a pressure gradient is applied, and the coefficient is backed out after flow rate is computed from first principles.)

For isotropic media, the pressure gradient can be applied in any of the Cartesian directions, and the Darcy velocity (specific discharge) will be aligned with the applied pressure gradient. However, the permeability of anisotropic media can vary with direction, and the full permeability tensor is required to characterize the transmissibility of such materials. In geologic materials, permeability can be anisotropic due to depositional effects and in-situ stress, and it often depends on the sample size and the resolution of the sampling method (Anderson 1994). In man-made materials, anisotropy can be introduced intentionally or as a byproduct of the manufacturing technique. In either case, quantifying anisotropy is an essential step in quantifying the material behavior in industrial applications.

### 2.2.2 Physically Representative Network Models

For image-based modeling, the use of direct numerical approximation of the equations of motion is appealing because it represents a first-principles approach. However, for real materials it typically requires very large computational overhead. A widely used alternative is network modeling, which is computationally efficient but less rigorous because of approximations made to both pore
structure and fluid mechanics. In the current application, we use network modeling because it is a reasonably quantitative approach for single-phase phenomena, and its speed makes it ideal as an analytical tool.

The basis of network modeling is to discretize the void space into a simplified geometry (a network of pores and pore throats), and then to impose mass conservation at each pore. This approach has been used to study porous media for over fifty years, and is attributed to Fatt (Fatt 1956). Physically-representative pore network (PN) models are models built on the pore structure of real materials. They have been used extensively over the last twenty years to characterize the void space in porous materials and perform efficient simulations for estimation of continuum properties and flow behavior. Bryant et al. (Bryant and Blunt 1992, Bryant, King et al. 1993, Bryant, Mellor et al. 1993) introduced PN models, which map the true pore structure of a material onto a network (unlike their predecessors which typically employed a lattice structure with computer-generated heterogeneity). Consequently, the PN models retain more of the true pore morphology and any inherent spatial correlations in the pore structure. The first networks of this type used an actual sphere packing that was experimentally mapped at the particle scale (Bryant, King et al. 1993). The technique was extended to computer-simulated materials, including porous media created to mimic sandstones via simple diagenetic processes (Bryant, Mellor et al. 1993) and packed-beds relevant to chemical engineering applications (Thompson and Fogler 1997). Other processes that have been studied include conductivity and permeability (Thovert, Salles et al. 1993), drainage capillary pressure and relative permeability (Bakke and Øren 1997), residence time distribution (Thompson and Fogler 1997), electrical conductivity and formation factor (Liang, Ioannidis et al. 2000), wormhole formation (Fredd and Fogler 1998), interfacial behavior (Prodanovic and Bryant 2006), and gas mobility in foams (Balan, Balhoff et al. 2011). In the past 10-15 years, the widespread availability of tomography has made image-based modeling the method of choice for many researchers. It has been applied to a variety of materials and processes including sandstones (Thompson, Willson et al. 2008), carbonates (Arns, Bauget et al. 2005), computation of permeability and scale (Fredrich, Digiovanni et al. 2006), relative permeability (Sheng, Thompson et al. 2011), multiphase flow (Ramstad, Oren et al. 2010), multiscale modeling (Sheng 2013, Sheng and Thompson 2013), and transport in fractures (Landry, Karpyn et al. 2014).

2.2.3 Characteristic Length of a Porous Material

Using 3D X-ray computed microtomographic (microCT) images to model fluid flow phenomena and predict properties of geologic samples has become possible due to improvements in both computational resources and image acquisition techniques. Still, as image resolution increases, larger computational domains are required to model the same size physical domain. Often, computational time and/or memory requirements become the limiting factor for the physical dimensions during image-based modeling, which risks the sample size being too small to represent the bulk material and quantitatively model its true properties.
Therefore, it is beneficial to be able to estimate the minimum characteristic length associated with a material sample prior to implementing computationally expensive simulations. Here, the terminology characteristic length means the representative length scale over which spatially averaged parameters can be found. It is expected to be different for different parameters and/or processes, and it contains no information about heterogeneity in the continuum-scale material. It is worth noting that the terminology Representative Elementary Volume (REV), which was originally defined in the volume-averaging literature, is commonly used to define this same concept: the length scale at which the continuum assumption becomes valid. We use the term characteristic length scale to emphasize that we make no claim as to whether the sample being modeled is either representative or elementary in the large-scale, volume-averaging sense. The characteristic length scale is defined as the length scale where one can cross the pore-scale to continuum-scale transition. It is equivalent to Bear’s description of REV (Bear 1972) (see Figure 2.1, where Bear defines REV as the volume where the property being studied converges).

Bear’s definition of REV defined using porosity states that the REV gives the minimum representation of the domain of the porous medium, and below which one must consider microscopic effects (individual pores). It is useful in the explanation of our objective. He stated that the dimensions of the REV are such that the effect of adding or subtracting one or several pores has no significant influence on the value of volumetric porosity. He also noted, however, that one may be required to define REVs of a medium on the basis of parameters other than porosity.

A study of the data-quality implications of REV estimation for porosity found that porosity-based REVs were sufficient for ensuring the reliability of interfacial area ($A_I$) and moisture saturation ($S_W$), even for systems in which $A_I$ and $S_W$ exhibited considerable heterogeneity (Costanza-Robinson, Estabrook et al. 2011). However, whether the porosity-based REV would successfully extend to systems containing a heterogeneous solid phase or would constrain other system properties to reliable scales of measurement was not concluded. It was later found that obtaining an REV for porosity is not sufficient to ensure that a satisfactory REV for permeability has been realized (Hendrick, Erdmann et al. 2012). The results of the current work agree with this finding and extend the idea to formation factor as well.
Based on the concept illustrated in Figure 2.1, a process for finding characteristic length is to begin with some sample volume, enlarge it incrementally while measuring the parameter of interest, and search for the point at which fluctuations in the estimated parameter die out. Since network model simulations can be carried out efficiently, one can use a reasonably large number of enlargements in order to observe the effects of changing the sub-network size. Before concluding this section we note that at least one other method has been developed for investigating the REV, rather than simply computing parameters for successively larger domains. Ovaysi et al. extracted the permeability REV based on flow patterns from pore-scale simulations, with correlation coefficients used to determine whether a given pore-scale sample was representative of flow at larger scales (Ovaysi, Wheeler et al. 2014). Mortar coupling was used to make this a computationally tractable problem.

2.3 Methodology

2.3.1 Porous Materials

Two types of porous materials were used in this study: computer-generated random sphere packs and real geologic materials imaged using x-ray microCT. Figure 2.2 shows an example of each type of material. They are described in more detail below.
Sphere packings are relatively simple and well-understood structures. Yet, they are good prototypes of many real materials. In this work we used computer generated packings containing 1000 and 10,000 spheres. They were generated using a simulated annealing algorithm with target porosities of approximately 38%. This algorithm operates using similar principles as the collective rearrangement algorithm described in Liu and Thompson (Liu and Thompson 2000), but it allows a broader set of constraints to be imposed if necessary. The 1000-sphere packing is shown in Figure 2.2. In the current paper, these packings are used to validate the basic network modeling code and then to create an anisotropic medium to validate the permeability tensor results.

The anisotropic sphere packing was created by stretching the 10,000-sphere packing along the x-direction. Specifically, the x-coordinate of each sphere was scaled by a factor of 1.2. All spheres remained the same size, hence the porosity increased from 38% to 48%. This process increases the permeability in all directions, but disproportionately in the y and z directions. The packing was then rotated in the xz plane by 30° so that the principle directions of permeability were no longer aligned with the Cartesian directions. The stretching process is pictured in Figure 2.3 (for a smaller packing). The full process is illustrated by Figure 2.4. It creates an anisotropic structure with known principal directions and permeabilities, thus allowing the tensor permeability algorithm to be validated.
Figure 2.3. Computer generated packings. Left: a random sphere packing containing 1000 spheres; Right: same packing with 100% stretching in x direction (x coordinate stretched, particles remain spherical)

Figure 2.4. Geometry operations for constructing a stretched and rotated packing from a 10,000 sphere randomly distributed packing.

Two geologic materials were also used in this work. Both were imaged using x-ray microtomography. The first is a dolomite sample with total porosity in the range 22% to 28% (different samples and different measurement methods) and an experimentally measured permeability of approximately 100 mD. The dolomite was imaged at a voxel resolution of 3.9 μm which is not sufficient to
obtain a high quality image of the pore space since porosimetry data and thin sections suggest that the intercrystalline pore throat dimensions are in the 5-10 μm range, with additional intracrystalline porosity in the submicron range. Nonetheless, the permeability from image-based modeling agrees well with experimental values (suggesting that the permeability is dominated by the larger pores), and it provides an interesting data set for demonstrating both the critical length algorithm and the permeability tensor algorithm. The second image is a microCT data set comprising a 0.36 x 0.36 x 0.522 cm³ section from a reservoir sandstone sample imaged at 3.52 μm/voxel. The experimental permeability is 2.0 Darcy. Additional details about this sample can be found in Sheng et al. (Sheng, Thompson et al. 2011)

2.3.2 Network Modeling

Once a microCT image has been segmented into solid and void phases, the void space is discretized into a network of pores and throats, and various geometric properties of importance are computed (location, inscribed radius, connectivity, surface area, etc.). This representation of the pore structure contains all information necessary to perform network modeling, from which one can compute continuum-scale properties of interest, such as effective porosity, formation factor, and permeability. Details on the methods used to extract PN models from digital data sets have been discussed previously (Al-Raoush, Thompson et al. 2003, Thompson, Willson et al. 2005) In our work, the algorithm operates by identifying the pore locations using maximal inscribed spheres, collecting voxels associated with each pore, and then computing network connectivity and geometric properties used in the network description. Figure 2.5 illustrates the network generation process for the dolomite image.

![Figure 2.5](image)

**Figure 2.5.** Left: a microCT image of dolomite with the pore network model superimposed. Right: The pore network model with pores and throats colored according to the pressure solution.
Computing porosity in PN models simply involves summing the volume of each pore. (Pore-throats are assigned conductance but not volume in our data structure.) While porosity can also be computed directly from image analysis (without creating a network), mapping it to the network allows it to be correlated with other parameters. For example, a related parameter, effective porosity, is important in the study of solute transport and subsurface flow. Effective porosity is the ratio of the volume available for fluid flow (i.e. the connected pore space between the specified inlet and outlet) to the volume of the sample. Isolated pore spaces and disconnected flow paths make no contribution to flow through the medium, but can impact plots of residence time distribution, for instance, if total pore volume is used. Effective porosity can also be computed easily from a PN model because the pore volumes and connectivities are known. Once inlets and outlets are specified, the volumes of all pores connected to at least one inlet pore and one outlet pore are summed and the total is divided by the volume of the domain.

Single-phase flow modeling is performed by applying a pressure gradient $\Delta P$ and solving the relevant conservation equations in the network. Specifically, pressure is specified at inlet and outlet faces of the domain, and the pressure field is computed by applying mass conservation at every pore. The flow through each throat connecting pore $i$ to pore $j$ is $q_{ij}$ given as:

$$q_{ij} = \frac{g_{ij}}{\mu}(p_j - p_i)$$

where $i$ represents all pores in the system, $j$ represents all neighboring pores connected to any pore $i$, and $p_i$ is the pressure in pore $i$. The throat hydraulic conductances, $g_{ij}$, are estimated using a set of parameters which characterize the physical dimensions of the local pore space and are typically derived from Hagen-Poiseuille-flow type solutions for capillary flows (although often for more complicated geometries than cylindrical). Because the capillary flow solutions are derived from the momentum equation, and the mass balance is imposed at every pore, the network model for flow can be viewed as a coarse numerical solution for the basic equations of motion. Many methods for calculating the $g_{ij}$ values have been published over the last several decades, and the algorithms described below can employ a variety of options. This allows the simulation to be tailored to some extent to the specific type of material being studied (Sheng 2013). Flow direction is specified and boundary conditions are applied as an imposed pressure drop. Inlet pores (those with throats connected to an inlet boundary) have a constant pressure and, outlet pores similarly have a different (lower) constant pressure. Prior to solution, pore throats connected to no-flow boundaries are removed, along with any dead pores which are not connected via a continuous pore space to both inlet and outlet boundaries. The following linear system of equations is then solved with a sparse matrix solver:
\[ \sum_{\text{pores}} \frac{g_{ij}}{\mu} (p_j - p_i) = 0 \]  

(4)

Once the pressure in each pore and flow through each throat have been determined, single-phase scalar permeability \( k \) is determined using Darcy’s law, just as is done with laboratory core-flood experiments. The total flow rate through the sample \( Q \) is found by summing the flow rates of all throats connected to the inlet, the sample size dictates the cross-sectional area \( A_{cs} \) and domain length \( L \), then \( k \) is computed for a given the simulated fluid viscosity \( \mu \).

\[ k = -\frac{Q \mu L}{A_{cs} \Delta P} \]  

(5)

Formation factor is another continuum-scale property used to quantify the electrical conductivity of a geologic material. Electrical conduction is governed by a Laplace Equation:

\[ \nabla^2 E = 0, \]  

(6)

where \( E \) is the local electric potential field. At the pore scale, the electric potential field is determined by solving for conservation of electrical current. Formation factor is defined as the ratio of the resistivity of a porous sample saturated with a conducting fluid (\( \sigma \)) to the resistivity of that fluid (\( \sigma_w \) for water):

\[ F = \frac{\sigma}{\sigma_w}. \]  

(7)

Resistivity is given by:

\[ \sigma = \frac{E_i - E_o}{I L A_{cs}}, \]  

(8)

where \( E_i \) and \( E_o \) are electric potential at the inlet and outlet, respectively, \( I \) is the electric flux between them. In a pore network, the pore space is mapped into an interconnected set of pores and pore throats. The governing equations are analogous to those for mass conservation:

\[ \sum I_{ij} = \sum g_{e,ij}(E_i - E_j) \]  

(9)

where \( I_{ij} \) is the electrical flux between pore \( i \) and pore \( j \), and \( g_{e,ij} \) is electrical conductance. Electrical conductance \( g_{e,ij} \) is estimated using a set of parameters that characterize the physical dimensions of the pore space. Bryant and Pallatt (Bryant and Pallatt 1996) defined the electrical conductance as:
\[ g_{e,ij} = \frac{A_{CS} \sigma_w}{L} \tag{10} \]

Like permeability, resistivity and formation factor are directionally dependent. Øren and Bakke (Øren and Bakke 2003) defined the average formation factor \( F \) as:

\[ \frac{1}{F} = \frac{1}{3} \left( \frac{1}{F_x} + \frac{1}{F_y} + \frac{1}{F_z} \right) \tag{11} \]

Geologic porous media are often comprised of minerals which are electrically insulating, in which case formation factor is always greater than unity.

Clearly, to compute directional permeabilities the flow direction can be controlled by specifying pressure boundary conditions on different (opposing) walls of the domain. As mentioned above, throats connected to the remaining walls are removed from the model along with all disconnected pores, as these make no contribution to the flow. Additionally, because the network provides knowledge of each pore’s location and connectivity to neighboring pores, one can extract smaller sub-domains from the network (from here, referred to as sub-networks). All pores and pore throats within a specified Cartesian region of interest (ROI) make up a sub-network and the same continuum-scale properties discussed above can be computed for each sub-network. This is the basis for determining the characteristic length, described next.

2.3.3 Characteristic Length Determination

An automated algorithm was developed to efficiently compute continuum properties for consecutively smaller sub-networks so that one can observe how these properties vary with domain size. In each iteration, the network model is cropped by a specified fraction and network simulations are run to determine effective porosity, permeability, and formation factor. These computations are performed by applying different boundary conditions for the three Cartesian directions, thus allowing directional parameters to be obtained if relevant. Characteristic length can then be inferred as the boundary-to-boundary size of the sub-networks at the point where the trends plateau (for homogeneous materials) or at least become smooth (but perhaps increasing or decreasing, for heterogeneous materials). Network model generation, which is the slow step, only occurs once, allowing the subsequent sub-network computations to be performed efficiently. Figure 2.6 illustrates the procedure by superimposing a volume rendering of a voxelized data set, along with the pressure solutions for successive sub-networks. Spheres and tubes are colored according to the network solution for pressure in the pores and throats, respectively.
As Bear (Bear 1972) described qualitatively, convergence of continuum properties with increasing sample size occurs once a characteristic length scale for that particular parameter is reached. Observing the trends of a property such as permeability for multiple flow directions simultaneously can provide additional insight into the anisotropy and/or heterogeneity of a material. If the principal flow directions are aligned with the coordinate system, this process also provides the permeability tensor. (If not, the procedure described in the following section must be used).

Figure 2.6. Illustration of characteristic length determination process. The network model is "cropped" in each iteration rather than the voxelized image.

2.3.4 Predicting the Permeability Tensor

In some situations, the anisotropic characteristics of a material are known prior to imaging (e.g., bedding planes in a geology material or fiber orientation in a fibrous material). This may allow sample alignment to be performed so that the Cartesian coordinates in the image data are aligned with principal direction(s) for permeability. In other cases, one may not have a-priori knowledge of whether a sample is isotropic and/or how the principal directions are oriented. In this latter case, the full permeability tensor cannot be determined simply by applying pressure drop in the three Cartesian directions as described previously. To overcome this limitation, we developed the following approach to efficiently compute this information using network modeling.

Three separate flow simulations are performed in three different (non-orthogonal) directions. Because the pressure gradients must be applied in directions that are not aligned with the Cartesian coordinates, the pressure cannot be applied uniformly on opposing faces, nor can any of the faces be specified as no-flow boundaries as is done in conventional network simulations. Instead, a single constant reference pressure is assigned to a boundary pore in the corner of the flow domain, and the imposed pressure gradient is
then used to assign pressure boundary conditions to boundary pores at all six faces of the network or sub-network. These boundary pressures are constant in time but vary spatially over the external faces of the domain as dictated by the reference pressure and the pressure gradient. Interior pressures are then computed in the same manner as with traditional network modeling. Typical results are shown in Figure 2.7. With pore pressures known, pore-scale flowrates are also known, and superficial velocity is computed by averaging the local pore-throat flowrates over a large number of interior pore throats that cross an oriented interior surface. Note that for anisotropic materials, the orientation of the macroscopic superficial velocity vector will not generally be aligned with the imposed pressure gradient. This process is performed for the three chosen directions mentioned above, at which point the permeability tensor can be computed as follows:

\[
\begin{bmatrix}
v_x & v'_x & v''_x \\
v_y & v'_y & v''_y \\
v_z & v'_z & v''_z \\
\end{bmatrix} =
\begin{bmatrix}
k_{xx} & k_{xy} & k_{xz} \\
k_{yx} & k_{yy} & k_{yz} \\
k_{zx} & k_{zy} & k_{zz} \\
\end{bmatrix}
\begin{bmatrix}
sin \phi_1 \sin \theta_1 & sin \phi_2 \sin \theta_2 & sin \phi_3 \sin \theta_3 \\
cos \phi_1 & cos \phi_2 & cos \phi_3 \\
sin \phi_1 \cos \theta_1 & sin \phi_2 \cos \theta_2 & sin \phi_3 \cos \theta_3 \\
\end{bmatrix}\left| \nabla P \right| 
\]

(12)

Figure 2.7. Network model colored according to pressure solution for flow in one of the three independent directions. Notice the gradient is not aligned with the Cartesian axes.
Here, $\mathbf{P}$ is the applied pressure gradient; $\phi$ is the angle between the pressure gradient and the $y$ axis; $\theta$ is the angle between the $z$ axis and the projection of the pressure gradient onto the $xz$ plane. Subscripts correspond to the three independent applied gradient directions.

$$X = \begin{bmatrix} \sin \phi_1 \sin \theta_1 & \sin \phi_2 \sin \theta_2 & \sin \phi_3 \sin \theta_3 \\ \cos \phi_1 & \cos \phi_2 & \cos \phi_3 \\ \sin \phi_1 \cos \theta_1 & \sin \phi_2 \cos \theta_2 & \sin \phi_3 \cos \theta_3 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}$$  \hspace{1cm} (13)

The elements of the permeability tensor, $k_{ij}$, are determined using equation 14, the notation for which is defined by equations 12 and 13.

$$A = ek - fh; \quad D = ch - bk; \quad G = bf - ce$$

$$B = fg - dk; \quad E = ak - cg; \quad H = cd - af$$

$$C = dh - eg; \quad F = gb - ah; \quad K = ae - bd$$

$$\det(X) = a(ek - fh) + b(fg - kd) + c(dh - eg) = aA + bB + cC$$  \hspace{1cm} (14)

$$\begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix} = \frac{1}{|\mathbf{P}| \det(X)} \begin{bmatrix} v_x & v'_x & v''_x \\ v_y & v'_y & v''_y \\ v_z & v'_z & v''_z \end{bmatrix} \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & K \end{bmatrix}$$  \hspace{1cm} (15)

Here, $v$, $v'$, and $v''$ are velocity elements corresponding to the three non-orthogonal gradient directions. Principal values of the permeability tensor are obtained from the eigenvalues of the tensor (Peters 2012). The principal axes for permeability of the material are given by the angles between the eigenvectors of the permeability tensor and the unit vectors of the coordinate system, where the angle $\theta$ between vectors $\mathbf{M}$ and $\mathbf{N}$ are computed as

$$\theta = \cos^{-1} \frac{\mathbf{M} \cdot \mathbf{N}}{|\mathbf{M}| |\mathbf{N}|}$$  \hspace{1cm} (17)

2.4 Results and Discussion

2.4.1 Network Modeling

While the characteristic-length and permeability tensor simulations will be applied mainly to microCT images of real materials, the use of a regular and/or well-characterized structure is important for validation. For validation of the basic single-phase network model, flow simulations were performed for two computer generated porous materials: (1) a random sphere packing containing 1000 spheres, which is expected to be isotropic; (2) a stretched (but not rotated) 1000-sphere packing; see Figure 2.3 and the earlier description. For these two packings, both network modeling and more rigorous computational fluid dynamics (CFD) simulations were performed to validate the network predictions. (The CFD simulations were performed using the finite element method with a
tetrahedral mesh of the pore space and the Stokes equations for low-Reynolds number flow solved using P2P1 elements (Lane 2011). Results are shown in Table 2.1.

Table 2.1: Permeability results for 1000 sphere packings using network model and FEM. (Permeabilities are reported without dimensions because sphere size is arbitrary in the computer-generated packings.)

<table>
<thead>
<tr>
<th></th>
<th>Network</th>
<th></th>
<th>FEM</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(k_x)</td>
<td>(k_y)</td>
<td>(k_z)</td>
<td>(k_x/k_z)</td>
<td>(k_x)</td>
<td>(k_y)</td>
<td>(k_z)</td>
<td>(k_x/k_z)</td>
</tr>
<tr>
<td>Random packing</td>
<td>9.84</td>
<td>9.58</td>
<td>9.32</td>
<td>1.06</td>
<td>9.80</td>
<td>9.73</td>
<td>8.96</td>
<td>1.09</td>
</tr>
<tr>
<td>Stretched packing</td>
<td>21.2</td>
<td>43.6</td>
<td>44.2</td>
<td>0.48</td>
<td>27.6</td>
<td>51.2</td>
<td>50.6</td>
<td>0.55</td>
</tr>
</tbody>
</table>

Results for the random packing are in agreement within 6%, which is not surprising because the network modeling code has been validated for this type of structure (Thompson and Fogler 1997). It is essentially isotropic, which is to be expected. For the stretched packing, permeabilities in all directions increase significantly, which is a consequence of the added void space between spheres. The increase is most pronounced in the two directions orthogonal to the stretch, which also is expected based on the change in the internal structure. For the stretched packing, the network permeabilities are in error by 15-30% relative to the CFD results depending on the flow direction. This difference is not unexpected since the same conductance formulas were used for a significantly higher-porosity material. The more important result from this set of simulations is that the network modeling is able to quantitatively capture the anisotropy that was present in the original packing structure, without any \textit{a posteriori} adjustments to the network. This result is shown by the permeability ratios in the stretched versus non-stretched directions, which agree to within 13%.

2.4.2 Characteristic Lengths of Porous Media

The algorithm developed for determining characteristic lengths of materials was applied to several data sets. Network simulations were carried out by the algorithm for flow in each direction (pressure gradients parallel to Cartesian \(x, y,\) and \(z\) axes). The sub-network domain dimensions were consecutively decreased towards the center of the original domain by 1% of the maximum dimension for 100 tests. Formation factor, effective porosity, and directional permeability were computed in each flow direction for each sub-network. The resulting data give insight about the isotropy and homogeneity of the sample and give confidence in the values of continuum properties which were computed (in cases where convergence was observed).

Networks were generated from the microCT images of the two real geologic materials previously mentioned (reservoir sandstone and dolomite). For non-cubic domains such as these, 10 additional iterations were carried out on sub-networks ranging from the largest cube-shaped domain to include the full data set in the long dimension (the last 10 data points on the plots; see Figure 2.10 for visualization of maximum cubic domain). The characteristic length algorithm was applied to both networks to generate the data in
Figure 2.8 for sandstone and Figure 2.9 for dolomite.

The sandstone data show that the sample was only slightly anisotropic. Effective porosity converged to about 30% at a sub-domain size of about 0.1 cm. By a domain length of 0.25 cm, the \( x \), \( y \), and \( z \) directional permeabilities converged to approximately 9.7\( \times 10^{-9} \), 8.2\( \times 10^{-9} \), and 10\( \times 10^{-9} \) cm\(^2\), respectively, before starting to smoothly increase with increasing domain size. This increase is assumed to reflect continuum-scale heterogeneity at lengths above 0.25 cm, but the smoothness of the trending data suggests the domain size is sufficient for capturing continuum-scale properties. The formation factor trends show a nearly mirrored image of the permeability results, which is consistent with the relationship between these two parameters. Before the effects of macro-scale heterogeneities were felt, the \( x \), \( y \), and \( z \) directional formation factor values were 13, 15.2, and 12.5, respectively.

The less permeable dolomite sample possesses a higher degree of anisotropy, with computed directional permeabilities varying from approximately 1.2\( \times 10^{-10} \) to 1.1\( \times 10^{-9} \) cm\(^2\) and formation factor from approximately 250 to 650. The data seem to be nearing convergence, but the persistence of fluctuations at sub-networks nearing the full domain size indicate that the sample size is not ideal for accurate determination of these properties. In other words, the characteristic lengths for formation factor and permeability are likely greater than 0.3 cm. Fluctuations in effective porosity remained near 11.5% for subdomains larger than 0.15 cm. (Although in principle, PN results should be relatively independent of pore density (Bhattad, Willson et al. 2011), it should be noted here that the sandstone network contained 233,433 pores whereas the Dolomite network contained only 8,009 pores.)
Figure 2.8. Characteristic Length Data for Reservoir Sandstone: Continuum Properties vs. Domain Length. Top: Formation Factor; Bottom: Permeability and Effective Porosity. Formation factor and permeability data shown for flow in x, y, and z directions. Only minor variations existed in effective porosity values between flow directions.
Figure 2.9. Characteristic Length Data for Dolomite
Figure 2.10. Largest cubic sub-network for dolomite. Volume rendering of microCT data occupies the remainder of the space between the max cubic sub-network and full-sized domain. This sub-network corresponds to the data points at $x = 0.203$ cm. Pores and throats are colored according to the pressure solution for flow in the positive $z$ direction. Black pores indicate disconnected void space which does not contribute to flow. The maximum sub-network (last point on the plot) would incorporate the pores seen embedded in the rock.

The critical-length algorithm was also run for the stretched and rotated packing described above, which contained 19,396 pores in the section extracted for simulation. Random sphere packings are expected to be homogeneous when a large enough domain is considered. Figure 2.11 presents the permeability and effective porosity trends. As expected, these continuum properties show obvious convergence to homogeneous values, but the three directional permeabilities are not equal. The stretching of sphere locations in the $x$ direction causes the $y$ and $z$ permeabilities to increase more than $x$, and the $30^\circ$ rotation about the $y$ axis makes the medium slightly less permeable in $z$ and more permeable in $x$. Two notes are the following: first, the reported permeability values are somewhat arbitrary because the packing is computer generated and thus sphere size can be selected arbitrarily; second, porosity is significantly larger than the original densely-packed value (38%) because of the stretching.
2.4.3 Permeability Tensor for Anisotropic Porous Media

The directional permeability results shown in Figure 2.8, Figure 2.9, and Figure 2.11 provide useful information. However, they are not sufficient to compute the principal flow directions, and thus may lead to inaccurate interpretation about material anisotropy. To characterize material anisotropy quantitatively, one must determine the full permeability tensor. To test the algorithm described above, the permeability tensor was computed for the stretched and rotated packing, which had a known anisotropic structure obtained from the geometric operations shown in Figure 2.4.

First, the permeability tensor was calculated for a random packing containing 10,000 spheres:

\[
\mathbf{k}_{\text{Random Pack}} = \begin{bmatrix}
1.47 & 0 & 0 \\
0 & 1.45 & 0 \\
0 & 0 & 1.46
\end{bmatrix} \times 10^{-7} \text{cm}^2
\]

The results indicate that the random packing is essentially isotropic. Next, the random packing was stretched by 20% in the \(x\) direction and a second tensor was computed. The stretched packing was then rotated by 30° and the permeability tensor of the new
packing was also determined. These geometric transformations are illustrated in Figure 2.4. The permeabilities computed for the transformed packs are:

\[
k_{\text{Stretched, Aligned}} = \begin{bmatrix} 3.16 & 0 & 0 \\ 0 & 4.40 & 0 \\ 0 & 0 & 4.44 \end{bmatrix} \times 10^{-7} \text{cm}^2
\]

\[
k_{\text{Stretched, Rotated}} = \begin{bmatrix} 3.55 & 0 & -0.61 \\ 0 & 4.47 & 0 \\ -0.61 & 0 & 4.19 \end{bmatrix} \times 10^{-7} \text{cm}^2
\]

Since the principal directions for the stretched packing are still aligned with the axes prior to rotation, the diagonal terms of the tenor are the eigenvalues and the off-diagonal terms are zero. Permeability values in y and z directions are expected to be nearly equal since the packing is stretched only in the x direction. After rotation, the principal values of the new permeability tensor were obtained from the eigenvalues, \( \lambda_i \), of the tensors: \( 3.18 \times 10^{-7} \text{cm}^2 \), \( 4.48 \times 10^{-7} \text{cm}^2 \) and \( 4.56 \times 10^{-7} \text{cm}^2 \). The off-diagonal terms in the tensor indicate that the pressure gradients will not be aligned with the Darcy velocity when applied in the Cartesian directions of the rotated data set. The agreement between the principal values before and after rotation validates the algorithm’s ability to recover principal permeability values when principal directions are not known a priori.

Angles between the eigenvectors \( (\nu_1, \nu_2, \text{ and } \nu_3) \) of the permeability tensor and the unit vector of the coordinate system \( (x, y, \text{ and } z) \) are listed in Table 2.2 along with the permeability corresponding to each eigenvector. The eigenvectors not aligned with the Cartesian axes were offset by approximately 30°, consistent with the 30° rotation of the stretched packing described previously.

Table 2.2: Angles between eigenvectors of the permeability tensor for the stretched, rotated random pack and the unit vectors of the coordinate system, and the corresponding eigenvalues (principal values)

<table>
<thead>
<tr>
<th>Eigenvector</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>Principal Value, ( \lambda_i ) (10^7 cm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nu_1 )</td>
<td>149°</td>
<td>90°</td>
<td>121°</td>
<td>3.18</td>
</tr>
<tr>
<td>( \nu_2 )</td>
<td>90°</td>
<td>0°</td>
<td>90°</td>
<td>4.48</td>
</tr>
<tr>
<td>( \nu_3 )</td>
<td>121°</td>
<td>90°</td>
<td>31°</td>
<td>4.56</td>
</tr>
</tbody>
</table>

The permeability tensors were also computed for the real porous media. The reservoir sandstone and dolomite samples were found to have the following permeability tensors, respectively:

\[
k_{\text{Sandstone}} = \begin{bmatrix} 1.18 & 0.05 & 0.044 \\ 0.05 & 0.995 & 0 \\ 0.044 & 0 & 1.136 \end{bmatrix} \times 10^{-8} \text{cm}^2
\]

\[
k_{\text{Dolomite}} = \begin{bmatrix} 0.50 & -0.03 & -0.06 \\ -0.03 & 0.84 & -0.02 \\ -0.06 & -0.02 & 0.48 \end{bmatrix} \times 10^{-9} \text{cm}^2
\]
Table 2.3 describes the principal flow directions by giving the angles between the eigenvectors \( \nu_1, \nu_2, \nu_3 \) and \( x, y, z \) Cartesian axes. The eigenvalues are also shown to complete the description of the anisotropic permeabilities. The results indicate that the dolomite sample is anisotropic, and the predicted principal directions for the permeability are not aligned with the computational block cut out of the microCT image. The sandstone was much more isotropic, and therefore the principal directions can be computed but are less meaningful.

One of the challenges with network modeling is choosing from the many possible formulas that have been proposed for computing \( g_\theta \) (Sheng 2013). However, additional numerical tests (not shown) indicate that predicted principal directions are nearly insensitive to the conductance computation provided it is a geometric-based computation appropriate for physically representative network models. This insensitivity is beneficial since it allows the anisotropy to be characterized more-or-less independently of the choice of conductance equation, which varies depending on the type of material or choice of the modeler.

Table 2.3: Angles between eigenvectors and the unit vector of the coordinate system for rock samples

<table>
<thead>
<tr>
<th>Sample</th>
<th>Eigenvector</th>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
<th>Principal Value, ( \lambda_i ) (( 10^{-9} ) cm(^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reservoir Sandstone</td>
<td>( \nu_1 )</td>
<td>75°</td>
<td>164°</td>
<td>94°</td>
<td>9.81</td>
</tr>
<tr>
<td></td>
<td>( \nu_2 )</td>
<td>64°</td>
<td>79°</td>
<td>151°</td>
<td>11.14</td>
</tr>
<tr>
<td></td>
<td>( \nu_3 )</td>
<td>31°</td>
<td>79°</td>
<td>62°</td>
<td>12.16</td>
</tr>
<tr>
<td>Dolomite</td>
<td>( \nu_1 )</td>
<td>50°</td>
<td>86°</td>
<td>41°</td>
<td>0.430</td>
</tr>
<tr>
<td></td>
<td>( \nu_2 )</td>
<td>41°</td>
<td>88°</td>
<td>130°</td>
<td>0.553</td>
</tr>
<tr>
<td></td>
<td>( \nu_3 )</td>
<td>94°</td>
<td>5°</td>
<td>92°</td>
<td>0.838</td>
</tr>
</tbody>
</table>

2.4.4 Speed of the Algorithms

The algorithms presented here are designed as analytical tools. They have particular value for rapid characterization and for preliminary analysis that is to help design more rigorous but computationally demanding simulations. Hence, speed is an important consideration. The computational requirements for the algorithms described in this paper are shown in Table 2.4. The critical length data require between a few seconds to 30 minutes for the networks shown here, while the permeability tensor algorithm requires at most two minutes. Memory requirements are modest, allowing all of these computations to be performed on a single processor.

The real value of network modeling when coupled with these algorithms is seen by considering that the characteristic-length-finder algorithm runs 330 network/boundary combinations (3 \( \times \) 100 cubic sub-networks plus 3 \( \times \) 10 additional sub-networks) and performs a permeability, resistivity, and effective porosity computation for each, totaling nearly 1000 data points. The computational time to acquire this amount of information via direct numerical simulation rather than network modeling would be prohibitive.
Table 2.4. Computational Requirements.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Number of Pores</th>
<th>Time (min:sec)</th>
<th>Memory (GB)</th>
<th>Time (min:sec)</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotated Random Packing</td>
<td>19,396</td>
<td>1:03</td>
<td>0.9</td>
<td>0:04</td>
<td>*</td>
</tr>
<tr>
<td>Reservoir Sandstone</td>
<td>233,433</td>
<td>31:50</td>
<td>18.7</td>
<td>2:06</td>
<td>0.7</td>
</tr>
<tr>
<td>Dolomite</td>
<td>8,009</td>
<td>0:12</td>
<td>*</td>
<td>0:04</td>
<td>*</td>
</tr>
</tbody>
</table>

* Less than 1 MB required

Architecture: 2.93 GHz quad core Nehalem Xeon 64-bit processors, 24Gb RAM available.

2.5 Conclusions

Two new algorithms were introduced in this paper that advance network modeling techniques for applied characterization of porous materials. The algorithms operate on high-resolution 3D images of porous materials, typically obtained from x-ray microtomography or a similar technique.

The first algorithm is used to compute critical characteristic length scales in the material for three important parameters: porosity, permeability, and resistivity. Quantitative results from the algorithm reflect the classic behavior that was predicted qualitatively by Bear (Bear 1972) when describing the concept of a representative elementary volume. Practically, the algorithm provides a way of determining the minimum characteristic scales for continuum-scale modeling of the material. It can also be used for rapid analysis of a digital image in order to determine the minimum-sized subsections that should be used for more computationally intensive image-based modeling techniques.

The second algorithm computes the full permeability tensor of a material using network modeling. This approach differs fundamentally from the common technique of determining permeability in the three Cartesian directions in that it allows one to extract the principal directions of permeability in an anisotropic material, regardless of whether these principle directions were aligned with the Cartesian directions in the image (or even known beforehand).

The time and memory requirements of both algorithms allow them to be run on a single processor with modest memory, thus allowing them to be performed on a variety of modern devices. Alternatively, as technology enables larger digital images (in terms of voxel dimensions), these approaches will allow modeling at larger length scales without being computationally prohibitive. This issue is particularly relevant to heterogeneous materials such as carbonate rocks and nanoporous materials for which the current pore-scale modeling may be limited to total domain sizes on the order of tens of microns.
Though tested for the three built-in parameters (permeability, effective porosity, and formation factor), the code is modular and externally callable, so that it can be easily adapted for use with other network modeling algorithms. A collaborator used the characteristic length code for species transport residence time distributions and found the length for that property to differ greatly from that of permeability (results not published). This is a useful tool to have alongside the hybrid multiscale framework described in the following chapter when doing any pore-scale modeling to ensure the results are valid within the bounding scales (i.e. enough microscale phenomena is captured via hybrid modeling to ensure accuracy, and a large enough domain is used for relevant continuum scale parameter prediction).
3 NETWORK-FEM MULTISCALE MODELING

3.1 Introduction

Flow in porous media is a research area that spans several industries and disciplines, including chemical production, hydrological and environmental studies, biomedical research, geothermal energy production, and petroleum reservoir engineering. Transport phenomena in porous materials is an inherently difficult problem to study due to the multiscale nature of many materials of interest to scientists and engineers. With different physical processes occurring at different scales within a medium and the interdependence of disparate scales’ phenomena on one another, the task of developing models which capture all relevant fluid transport behaviors quickly becomes quite challenging.

Despite the difficulties associated with the complex pore geometries and heterogeneities found in many porous media, there are modeling techniques which can help to shed light on the physical processes occurring on each scale. These include fundamental approaches, such as the direct solution of either the equations of motion in the pore space as in the finite element method (FEM) or collision models as in the lattice-Boltzmann method (LBM). Alternative, and less rigorous, techniques such as pore network (PN) modeling are also available and are markedly more computationally efficient, allowing flow to be modeled over orders-of-magnitude larger physical domains. While the PN was the most popular tool for pore-scale modeling in the past, LBM became prevalent more recently because of its parallelizability and the ease of implementing it on voxelized images such as tomography data, along with the advances in modern computing hardware. Still, LBM is computationally too expensive for high resolution simulation of large domains. Though often avoided due to past limitations concerning computational time, memory, and meshing capabilities, FEM also offers significant advantages because of its ability to operate in multiscale and multiphysics frameworks. Perhaps the greatest benefit associated with using FEM is its ability to use an unstructured discretization of the pore space. This allows adaptive element sizing and gives the model the ability to handle complicated geometries quite well. Unlike LBM, which uses each voxel as a volume element on a uniform grid, FEM meshes are independent of image resolution. Therefore, high-resolution images afford FEM models better representations of pore structure without incurring prohibitive computational requirements.

3.2 Background

3.2.1 Length Scales

The study of porous materials is nearly always a multiscale endeavor, where most practical applications span several to many orders of magnitude. Packed bed reactors in chemical manufacturing applications (Thompson and Fogler 1997, Dadvar and Sahimi 2003, van der Hoef, van Sint Annaland et al. 2008), polymer-electrolyte fuel cells (Gostick, Ioannidis et al. 2007, Medici, Zenyuk et al. 2016), manufacturing of fibrous and composite materials (Hou and Wu 1997, Thompson 2002), groundwater transport and
decontamination (He and Ren 2005, Bear and Cheng 2010), petroleum production reservoirs (Aarnes, Kippe et al. 2007) and geothermal energy production reservoirs (Revil, Schwaeger et al. 1999, Blöcher, Zimmermann et al. 2010) are all examples of porous media flow problems where phenomena occurring on multiple length scales must be captured to completely understand, predict, and/or control the process.

The scales for porous media models vary from streamline to field scale, and are illustrated in Figure 3.1. In macroscale problems, the void and solid phases are not separated. Rather, the entire medium is treated as a continuum, where properties such as porosity and permeability are generally constant over each region. At the pore-scale level and below, discrete models are used where the physical location of each phase in the material (solid, void, oil, water, etc.), and interfaces between them are known. As pore-scale modeling techniques have evolved, it has become necessary to distinguish between different approaches, particularly the differences between true pore-scale models versus streamline-scale models. *Pore-scale* often refers to all discrete models, but we will make a distinction here between *pore-scale* and *sub-pore-scale*. *Pore-scale* will denote the highest level of discrete models, where the characteristic scale of discretization is on the same order as the pore size. Pore size varies between materials and can span orders of magnitude for a single material in some cases, but state variables such as pressure and phase saturation do not vary spatially within an individual pore. *Sub-pore-scale* and *streamline-scale* will be used when referring to models where the problem is discretized at a level finer than the pore size and pressure, velocity, and concentration fields are resolved within individual pores. Using this distinction, pore network models are the only models commonly applied to the pore-scale. A variety of other discrete models would fall into the sub-pore-scale category. These include the conventional CFD/DNS methods (Computational Fluid Dynamics / Direct Numerical Simulation: Finite Element Method (FEM), Finite Volume Method (FVM), etc.), Lattice Boltzmann Method (LBM), Meshfree methods (Smooth Particle Hydrodynamics (SPH)), and many others.
Mehmani and Balhoff (2015) describe pore network models as “intermediate-scale”, or “mesoscale” models which bridge the gap between the pore and core scales. This is an appropriate category for these models and is helpful in describing the discrete-to-continuum hybrid models used in that work.

The current work is focused on image-based modeling at the pore-scale and sub-pore-scale. The term image-based modeling suggests that a discrete model is being implemented, where a 3-dimensional image was used to identify the locations of the discrete phases. Images used in this study were provided by collaborators or acquired from an open online repository (Prodanovic, Esteva et al. 2015). Therefore, an in-depth discussion on X-ray micro computed tomography imaging (X-CT, or CT for computed tomography), image analysis, and image segmentation will not be presented here. A brief overview is provided in section 3.3.1, and a recent review on the state-of-the-art of image-based modeling has been given in (Bultreys, De Boever et al. 2016).

Recent trends in porous media modeling show increasing interest in hybrid and/or coupled models on multiple length scales. (See Figure 3.2 for trends in porous media modeling literature at various scales.) Most recent multiscale porous media literature involves bridging the gap between the pore-scale and Darcy-scale, coupling discrete and continuum models. This is an important area for research because the domain sizes of most practical problems of interest are above the limit of computational tractability for full discrete models, and full continuum models cannot adequately capture the microscopic phenomena that contribute to essential macroscopic properties/modeling parameters.
Figure 3.2. Yearly trends in network model publications from Web of Science data (www.webofknowledge.com 2017). 2017 shown as empty circles to indicate incomplete data. Top: trends of papers published on pore-scale, continuum-scale, and reservoir-scale models. Bottom: trends of multiscale and total network modeling papers suggest approximately 40% of network modeling papers in the last 5 years were associated with multiscale problems.

3.2.2 Physically-Representative Pore Network Models

The general approach of network modeling is to discretize the void space into a more simplified geometry (a network of pores and throats), then impose a mass conservation equation at each pore. This approach has been used to study porous media for over fifty years, and is first attributed to Fatt (Fatt 1956), though most models used today are much more sophisticated and useful than the early lattice-based networks employed at that time. Pore network models (PN) have been used extensively over the last 20+ years to characterize the void space in porous materials and perform efficient simulations for rapid estimation of continuum properties and pressure distribution. Bryant et al (Bryant, Mellor et al. 1993) introduced a physically-representative PN which mapped the true pore structure of a material onto a network, unlike predecessors which typically employed an underlying lattice. Consequently, this model
retained the true pore morphology and any inherent spatial correlations more completely. The networks were used to compute permeabilities for artificial porous media which were generated by simulating the typical processes by which sandstones form (Bryant, Mellor et al. 1993) and were similarly used to model a real random sphere-packing for which the geometry and topology had been completely characterized (Bryant, King et al. 1993). Soon after these initial papers, a number of other groups developed models in which the network structure is derived directly from a well-characterized material, rather than derived from statistics. Thompson and Fogler (1997) used physically-representative PNs to study flow in simulated, disordered, packed beds and developed a more general model for fluid and solute transport through such porous media. Several researchers began extracting PNs from tomographic images of real porous media in order to compute conductivity and permeability (Thovert, Salles et al. 1993), and to predict drainage capillary pressure and relative permeability (Bakke and Øren 1997) of geologic materials. Since their introduction, PNs have been regularly used for rapid estimation of many other single- and multi-phase properties including permeability (Bryant, King et al. 1993), residence time distribution (Thompson and Fogler 1997), relative permeability, interfacial area (Øren, Bakke et al. 1998), effective viscosity for non-Newtonian flows (Lopez, Valvatne et al. 2003), dispersion coefficients (Alvarado, Davis et al. 1997), electrical conductivity and formation factor (Liang, Ioannidis et al. 2000), as well as to study wormhole formation (Fredd and Fogler 1998). In the last 15 years, there has been a focus on increasing the realism of the pore-scale physics of network models (Blunt, Bijeljic et al. 2013), which is an objective of the current study.

Due to the vast amount of literature on pore network modeling, it is important to make the distinction that in this work the characterization of the pore space is a pore body/throat network and not a pore body/channel network. (i.e. each throat is defined as a surface of contact between two pores, rather than a channel with finite volume connecting two pores.). These throat surfaces are not necessarily planar, but they occupy no volume. Therefore, all the volume of the void space is contained in the pores. In this study, an in-house code named Vox2Net is used to generate PNs from either (segmented) voxelized images of real porous media or simulated geometric packings (Thompson, Willson et al. 2005, Bhattad, Willson et al. 2011). The algorithm is discussed further in section 3.2.2.1.

3.2.2.1 Pore Network Extraction Methods

Bhattad, Willson et al. (2011) provided an overview of some network generation techniques, along with an illuminating account of the effects of image-based network generation techniques on flow modeling. They tested four network generation methods on three types of porous materials. Each method yielded vastly different networks for each material in terms of pore-size distribution and interconnectivity, while maintaining the same total pore volume and surface area as the voxel image. The main takeaway from the paper is that single-phase permeability for each material was relatively insensitive to the network generation method used. The
authors stated that no single network model (out of the techniques tested in the study) is uniquely correct, except in cases of structured sphere packings. That was in the case of permeability (continuum-scale) calculations. For the current work, which focuses on pore-level phenomena, the pore density (and by extension, throat shapes/sizes) does impact the hybrid model, as discussed in section 3.2.6.5.2.

The sophistication of network model extraction techniques has drastically evolved over the years. There are many nuances and alterations that different research groups have made while building off previous techniques. This evolution can be roughly grouped into three generations of network models (Bhattad, Willson et al. 2011).

The original network models, developed by Fatt (1956), were simple two-dimensional networks of tubes arranged on square and hexagonal lattices, where volume-less pores were defined as intersections of tubes. Tube sizes were assigned randomly such that the distribution matched an experimental pore-size distribution. Chatzis and Dullien (Chatzis and Dullien 1977) extended the technique to three dimensions and multiple fluid phases. They found significant differences in the results of two- and three-dimensional networks, especially for immiscible displacement. The next generation of extraction techniques involves mapping a network directly from the void structure of a 3d packing. Bryant (Bryant, Mellor et al. 1993, Bryant and Johnson 2003) introduced this method as an improvement over using a lattice structure. Third-generation networks are mapped directly from high-resolution digital images of porous media (Thovert, Salles et al. 1993, Lindquist, Lee et al. 1996, Delerue, Perrier et al. 1999).


The current work utilizes a third-generation extraction technique. Details on the method used to extract physically realistic pore network models from digital data sets have been discussed in previous publications (Al-Raoush, Thompson et al. 2003, Thompson, Willson et al. 2005, Bhattad, Willson et al. 2011). In summary, the algorithm operates by identifying the pore locations using maximal inscribed spheres, collecting voxels associated with each pore, and then computing network connectivity and geometric properties used in the network description. When collecting the voxels which make up each pore, a labeled voxel image referred to throughout this work as the poremap is generated. This 3D label field is used throughout the multiscale modeling workflow. Figure 3.3 shows a poremap and the corresponding PN. Options available for tailoring the PN to realistically characterize the structure and
flow dynamics of the porous material include choices of initial pore seeding locations and pore merging criteria. PNs used in this study were all extracted using voxel burn maxima, which provide approximate pore locations, as seeds for potential pore centers. Different merging options were tested in this study and are discussed in 3.3.4.1.

3.2.2.2 Single-Phase Network Modeling

A single-phase flow solver developed by Sheng (2013) is used to efficiently estimate permeability, and pressure and flow distributions from PNs. The fundamental physical laws of momentum and mass conservation at the pore scale are the Navier-Stokes (18) and continuity (19) equations, respectively.

\[ \rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\nabla p + \mu \nabla^2 u \quad (18) \]
\[ \nabla \cdot u = 0 \quad (19) \]

\( \rho \) is fluid density, \( u \) is velocity, \( p \) is pressure, \( \mu \) is viscosity, and \( t \) is time. For low-Reynolds-number flow of a Newtonian fluid, the governing equation for flow \( q_{ij} \) through throat \( ij \), connecting pores \( i \) and \( j \) is

\[ q_{ij} = g_{ij} \frac{\mu}{(p_j - p_i)}. \quad (20) \]

The throat’s absolute hydraulic conductance, \( g_{ij} \), is estimated using a set of parameters which characterize the physical dimensions of the local pore space and is typically derived from Navier-Stokes via simplifications in both fluid dynamics and pore geometry.

Many methods for calculating the \( g_{ij} \) values have been published over the last several decades, and a detailed discussion on hydraulic conductivity models follows in section 3.2.2.3. The most common and earliest method is an assumption of cylindrical throat geometries where the Hagen-Poiseuille equation (21) applies. The throat radius \( R_{ij} \) and length \( L_{ij} \) can be defined in various ways (Blunt and King 1990, Bryant, Mellor et al. 1993, Mani and Mohanty 1999, Joekar-Niasar, Hassanizadeh et al. 2008).
\[ g_{ij} = \frac{\pi R_i^4}{8 L_{ij}} \] (21)

Flow direction is chosen, and boundary conditions are set as a constant applied pressure gradient. Inlet pores (those with throats connected to an inlet boundary) have a constant pressure and, outlet pores similarly have a different (lower) constant pressure.

Pore throats connected to no-flow boundaries are removed, along with any dead pores which are not connected via a continuous pore space to both inlet and outlet boundaries. Then, mass conservation in each pore is enforced and the sparse linear system of equations (22) is solved with a library sparse solver.

The direct result is the pressure distribution throughout the network (pressure of every pore), which can be used to calculate flowrates using the assumed hydraulic conductivity expression. From this, the total flow through the material is known which allows

\[ \sum_{\text{all pores } i \text{ and neighbor pores } j} q_{i,j} = 0 \] (22)

us to compute permeability. A ball-and-stick representation of a PN extracted from a random sphere pack is shown at the bottom of Figure 3.3, with the pores and throats colored according to the pressure solution.
Figure 3.3. Physically-representative pore network extracted from random sphere pack. Left: Poremap with voxels labeled, solid phase in grey. Right: Ball-and-stick representation of PN, colored according to single-phase flow solution pressures (Red-high, Blue-low)
3.2.2.3 Hydraulic Conductivity Estimation Methods

There have been many different approaches to estimating the hydraulic conductivities of throats in physically-representative pore network models. The following is an attempt to provide a comprehensive review of the ways network modeling enthusiasts have estimated single-phase throat conductance. A companion table of equations is provided in APPENDIX D.

Considering the various ways researchers discretize pore spaces into networks (pore body/channel network, where pores are volume-less nodes connected by channels with finite volume; Bryant and Blunt 1992, Bryant, King et al. 1993); pore body/throat network, where throats have no volume and are defined as contact surfaces between volume-containing pores (Bakke and Øren 1997, Thompson, Willson et al. 2008); combinations where volume is distributed to both pores and throats (Silin and Patzek 2006, Dong and Blunt 2009), naturally there are conductivity methods which do not apply to every network. For this reason, it was impractical to compare all known methods for computing hydraulic conductivity using the one extraction algorithm. Unfortunately, it is also likely that several techniques have not been represented here, considering the massive amount of literature dealing with network models, across various disciplines and spanning over 50 years. This compilation of methods is a continuation of the work of Sheng (Sheng 2013).

For lattice-based PNs, hydraulic conductivities can be determined statistically by sampling from a distribution of observed or computed values (Celia, Rajaram et al. 1993). One can assume a strong correlation between conductance and adjacent pore volumes, and sample from the log-normal distribution of log-volume summation (Li, Peters et al. 2006). For predictive, physically-representative PNs, we are interested in capturing pore-level phenomena, not average macroscopic effects. Bryant (Bryant, King et al. 1993) noted that the accuracy of permeability predictions does not necessarily establish the validity of assumptions used in PN calculations because the amount of information lost by the drastic assumptions and fundamental errors could conceivably cancel each other. Since permeability is an average macroscopic property, using any method for computing conductance will result in accurate estimation of permeability if an adjustable tuning parameter is included. Therefore, while permeability comparison is reported throughout literature as model validation, it is not an adequate measure of network model accuracy and pore-scale flow distributions must be considered when validating predictive network models.

3.2.2.3.1 Throat cross section geometry assumptions

A common approach to estimating the hydraulic conductivity of a throat is assuming all throats have some geometry for which an analytical or empirical Navier-Stokes solution exists, and mapping the properties of the throat onto that geometry. Assigning the same shape to all throats in the network simplifies calculations, increasing efficiency and ease of implementation at the expense of accuracy. This was the approach used in the earliest PNs (Fatt 1956) and the earlier bundle of tubes approach (Childs and Collis-
George 1950), where the entire porous matrix was represented as interconnected or parallel circular tubes. When circular tube geometry is assumed, the Hagen Poiseuille equation (21) is used. Ewing and Gupta (1993) used the equation for flow through a circular orifice (Happel and Brenner 2012) – a one parameter model in which the throat conductance is a cubic function of throat radius, ignoring the effect of throat length, and variation of throat radius along the flow axis. White (1991) provided exact solutions for Poiseuille flow in some noncircular ducts, which have since been used in conduit flow calculations (Lewis and Boose 1995) and were later used for computing hydraulic conductivities in porous media network models (Lock, Jing et al. 2002, Bijeljic and Blunt 2007). Analytical conductance solutions have been used for elliptical (Koplik, Lin et al. 1984, Seeburger and Nur 1984), rectangular (Perrin, Tardy et al. 2006), triangular (Jia, Dong et al. 2008), and less familiar throat cross sectional shapes such as the four-cusp (Goode and Ramakrishnan 1993).

Bryant (Bryant, King et al. 1993) divided void spaces into small segments with conical frustum geometry, then estimated the conductance for each segment, and used the harmonic mean of segment conductances for the total conductance. Similarly, the harmonic mean of the conductance of two neighboring pores and connecting throats has been used, where throat conductance was a tabulated solution for rectangular cross-section and pore conductance was a cubic function of pore radius (Ioannidis and Chatzis 1993), Poiseuille solution through cube-shaped throats and pores (Matthews, Moss et al. 1993), shape factor-corrected Poiseuille solution (Bakke and Øren 1997), or Poiseuille solution using effective radii (Mogensen and Stenby 1998). Sholokhova, Kim et al. (2009) followed the shape-factor based series resistance method of Bakke and Øren, using a weighted mean of the conductances of two pores and their connecting throats. They studied the sensitivity of predicted macroscopic permeability to the weighting factors.

Koplik developed a ‘ball and stick’ model (Koplik 1982, Jerauld and Salter 1990), which accounted for the radius variation along the flow axis by adding the contributions from throat entrance and exit effects into the Hagen-Poiseuille conductance expression. Thompson and Fogler (1997) derived a Poiseuille flow correction term, the ratio of resistance to flow through a venturi to that of a circular tube. Similarly, Schlueter (1995) corrected the overestimation of assuming straight tubes by defining a constriction factor to reflect the extent to which the radius varies along the path axis. Using the constriction factor, she derived an analytical solution for circular cross sections in which the variation of radius is sinusoidal. Lock, Jing et al. (2002) derived the constriction factor for sawtooth-varying tubes. Sisavath, Jing et al. (2001) derived a conductance formula using an asymptotic series solution for sinusoidally varying throats. They found the asymptotic series solution was more accurate than Hagen-Poiseuille lubrication solution for constricted throats, which does not account for the wavelength of constrictions. Øren and Bakke (2003) used effective pore and throat lengths to calculate the conductance in two pores and their connecting throat. (Dong and Blunt 2009) made a similar correction, but used different weighting factors to calculate effective lengths. Throat length is often computed as the distance...
between the two adjacent pore centers. Bryant, King et al. (1993) stated that several throats are connected to a pore, this can cause a portion of the flow path to be counted multiple times, overestimating the distances traversed by elements of fluid and thus systematically underestimating conductances. To correct this, they proposed approximating throats as cylindrical tubes and shortening the lengths of flow paths according to the average length of overlap for each tube.

3.2.2.3.2 Shape factor based formulas

Mason and Morrow (1991) defined the dimensionless shape factor $G$ as the ratio of area to the square of perimeter:

$$G = \frac{A}{P^2} \quad (23)$$

This parameter is also known as a dimensionless hydraulic radius (Patzek and Silin 2001), and it is useful for characterizing the shape of a cross section. Values near zero correspond to slivers, while equilateral triangles and circles have shape factors of about 0.048 and 0.08 (the maximum value of $G$), respectively. Combined with the aforementioned exact solutions for noncircular conduits (White 1991, Patzek 2000), shape factor can be used to choose which geometry a throat should be mapped onto and which analytic equation should be used (Patzek and Silin 2001). The shapes of pore cross sections are represented by a simple shape with a $G$ that matches that of the real pore shape. The conductance can then be computed by assuming a linear relationship between $g_{ij}$ and $G$ (Øren, Bakke et al. 1998, Valvatne and Blunt 2004), or computing a shape-based correction factor to the Poiseuille equation (Mortensen, Okkels et al. 2005). Mortensen, Okkels et al. (2005) also computed corrections for harmonically-perturbed circular cross sections. Ryazanov et al. (Helland, Ryazanov et al. 2008, Ryazanov, Van Dijke et al. 2009) noted that C-T-S (circle, triangle, square) shape characterizations suffer from being purely convex, which is not representative of throat cross sections commonly observed in images of real materials. In contrast, they represented throats as regular, $n$-cornered star shapes. Joekar-Niasar (2010) referred to C-T-S models as “mixed cross-sectional” networks, and noted that in previous shape factor-based formulations in literature, the distributions of shape factors in the networks were stepped since throats were binned according to the nearest arbitrary regular polygonal shape. The study also developed a procedure for selecting cross-sectional shapes of pore elements based on shape factors and image analysis data; throats were represented by tubes with $n$-hyperbolic polygonal cross sections ($n$ denotes the number of vertices; cross sections based on irregular hyperbolic triangles or regular hyperbolic polygons) in an attempt to recover a continuous distribution of $G$. Sholokhova, Kim et al. (2009) found that models employing channel conductances based on shape factor did not capture a sufficiently rapid increase in conductance with increasing shape factor.
3.2.3.3 Empirical and CFD-derived formulas

Thompson and Fogler (1997) used FEM solutions of parallel flow through ducts with the same cross sections as pore throats to determine each throat’s effective radius \( R_{CFD} \) for mapping the actual pore geometries onto a hyperbolic venturi and computing a correction factor to the Hagen-Poiseuille equation. The context of that study was packed beds, and Delauney tessellation of spheres was used to determine throat cross sections.

Sholokhova, Kim et al. (2009) also developed a hybrid PN using LBM-based channel conductances. Small, efficient LBM simulations were conducted for each throat, with boundaries being the centers of mass for each pore or 37 micron on each side of the throat surface (whichever is shorter) so that the CPU resources required were cheaper than full LBM simulation of the domain. They also derived empirical conductance formula based on LBM results. This was a similar idea to the multiscale approach proposed in this study, as information from streamline-scale simulations was used to improve the pore-scale model. However, that work used a pore body/channel network (as opposed to the pore body/throat network in this work), and the arbitrary, unrealistic boundary conditions applied to each channel LBM simulation, while allowing easy implementation and minimizing computational expense per throat, differ from the pore-cluster subdomain approach which will be proposed in the following sections.

3.2.3 Finite Element Stokes Flow Modeling

The finite element method has been a common approach for the modeling of mechanics and fluid flow problems for decades. The approach has been less common in the field of porous media, however, due to meshing challenges and computational requirements associated with the geometric complexity and the need to incorporate boundary conditions via multiscale techniques. Only a small number of pore-scale FEM studies in the field of porous media flow exist in literature to the knowledge of the author. Most of these studies were performed on simple 2D geometries or packings of spherical particles in 3D (Snyder and Stewart 1966, Ghaddar 1995, Zhu, Fox et al. 1999, Fourar, Radilla et al. 2004, Akanji and Matthai 2010). Akanji, Nasr et al. (2013) used CAD-based unstructured Stokes FEM formulation to compute permeability and studied the effect of particle shape and anisotropy using unconsolidated granular packs. Bird, Butler et al. (2014) used the Avizo and COMSOL commercial software packages for 3D Stokes flow simulation in small CT images Fountainebleau sandstone and a marly dolostone.

Lane and Thompson (2010) modeled single phase Stokes flow through Berea sandstone using an image-based, unstructured mesh generation technique which was developed in-house, specifically for porous media meshing. These meshing and simulation algorithms allowed the study of flow in proppant packs (Sanematsu, Shen et al. 2015), nanoparticle transport in porous media (Molnar, Sanematsu et al. 2016), inertial flows in porous media (Shen 2014), and hydrodynamic drag forces on colloidal particles in...
porous media (Farahani 2014). Borujeni, Lane et al. (2013) studied the effects of image resolution and numerical resolution on computed permeability, and compared LBM and FEM simulations of flow through computer-generated consolidated porous media. These works opened the door for this study.

The discretization of the pore space prior to FEM simulations is a major contributing factor to the accuracy of the numerical solution (Borujeni, Lane et al. 2013). While meshes of extremely high resolution (very many elements per pore) can be computationally prohibitive, mesh quality and conformation to the underlying medium are factors which must be considered to obtain suitable results. The in-house tetrahedral mesh generation code used in this work is described in further detail in section 3.2.3.1.

For incompressible Newtonian fluids, the Navier-Stokes equation \((18)\) describes the velocity field, \(u\), of continuum fluid substances, accounting for both inertial and viscous effects. The two terms on the right side are the pressure and viscous terms, and the unsteady and convective terms on the left side describe the effects of inertia. As the Reynolds number approaches zero, which is common for porous media flow, inertial effects become negligible and the flow behavior can be accurately described by the Stokes equation. This regime is known as creeping flow, where the inertial terms of the Navier-Stokes equations (left-hand side of equation \((18)\)) can be neglected. The Stokes equations, like Darcy’s law, describe a linear relationship between pressure drop and velocity.

The Stokes and continuity equations are shown below with the incompressibility constraint (density invariant).

\[
\nabla P = \mu \nabla^2 u \tag{24}
\]

\[
\nabla \cdot u = 0 \tag{19}
\]

It is advantageous to compute the velocity and pressure fields of fluids in porous media to attain a deeper understanding of local microscopic flow phenomena, and so that accurate flow rate/distributions can be determined as well as useful quantities such as drag force. The FEM formulation used here solves the Stokes equation with a Bubnov-Galerkin scheme using P2P1 (Taylor-Hood) elements, meaning pressure is approximated linearly while velocity is approximated quadratically. Taylor-hood elements provides a continuous pressure approximation, better accuracy, and cheaper computation than many discontinuous pressure elements, but have the unfortunate disadvantage that they do not explicitly enforce mass conservation at the element-level. This issue is discussed in detail in APPENDIX A.

### 3.2.3.1 Subdomain Extraction

One of the difficulties encountered when coupling models at different scales is that communicating information between models at boundaries is complicated by differences in discretization. Most image-based models are based on the cartesian coordinate system and boundary conditions are imposed along the outer faces of the domain. Apart from simple cubic sphere packs, interior pore
spaces do not naturally conform to the coordinate axes, so selecting a rectangular interior subdomain for sub-pore-scale modeling is an arbitrary choice which necessitates additional assumptions to be made during the coupling process. To avoid this, a flexible subdomain extraction algorithm was developed which selects clusters of connected pores based on the poremap. The rationale behind it was to assist in identifying equivalent subdomains in the PN and the voxel image, and to assist in meshing subdomains which correspond to a particular set of pores in the PN. If a known region of interest is to be meshed, the bounding coordinates are specified, and the extracted subdomain consists of the voxels of the pores of interest (POI) – those fully-contained within the region. Due to the variation in pore shapes and sizes, the cluster of extracted voxels to be meshed will not be cuboid. The interface between the voxels of two pores becomes a subdomain boundary if only one pore is a POI. The pressures of all pores (POI and external) are computed by the PN simulation, so pressures can be interpolated at boundary throats and applied as Neumann BCs (constant normal tractions equal to the applied pressures). These boundary conditions will reflect the larger-scale heterogeneities and flow profile of the full domain. This results in a more physically-realistic flow and pressure profile within the FEM model, and is a vast improvement over specifying two arbitrary constant inlet and outlet normal tractions at two faces of a Cartesian subdomain and no-slip conditions at the four lateral sides of the boundary, as has been done in the past.

3.2.3.2 Unstructured Mesh Generation

Unstructured meshes are well suited for pore-scale modeling for their ability to conform to complicated geometries and local spatial refinement. Unlike structured discretizations, the numerical resolution of unstructured meshes is not tied to the image resolution in for image-based modeling (Borujeni, Lane et al. 2013), which is a significant advantage over the popular grid-based approach of LBM.

The implicit voxel-based meshing strategy used in this work consists of multiple algorithms run in sequence with many adjustable parameters to control mesh resolution, point distribution, node and element labeling, and other properties of the mesh. The following procedure was used for discretizing a subdomain of the pore space into an FEM mesh which is to be coupled to a network model.

1. Once the segmented 3D voxel image is used to generate the PN and the poremap as discussed in previous sections, the pores of interest are identified and the subdomain to be meshed is specified. Subdomains should be selected so that no meshed pores are in contact with the outer image boundary so that more realistic boundary conditions from the PN solution can be applied, thereby minimizing boundary effects and avoiding the application of arbitrary constant pressure (traction) to all nodes on the boundary face.
2. Points are generated and randomly distributed throughout the domain of interest by randomly packing spheres and taking the sphere centers as the set of points to be tessellated. This allows a great deal of flexibility in varying the resolution of the mesh as the process of inserting points is totally independent of the voxel size. Regions of refinement can be specified where denser point distributions are necessary, typically near solid boundaries and in tight pore constrictions.

3. A Delaunay Tessellation of the points is carried out to create the tetrahedral elements of the mesh.

4. Elements intersected by the interfaces between pores of interest and the solid phase (or pores not of interest) are identified and subdivided, with new nodes placed at the point of intersection.

5. All elements remaining in the pores of interest are then collected and can be smoothed to achieve better quality elements (to meet specific criteria for aspect ratios and/or condition numbers (Freitag and Knupp 2002) of elements, or pore volume, surface area and/or surface conformation of the mesh overall).

6. Elements and nodes are then labeled with the appropriate integer pore label according to the pore label at that location in the poremap (voxel image).

Examples of labeled meshes are shown for a typical cuboid domain (Figure 3.4) and new pore-cluster subdomain (Figure 3.5). Lane (2011) provided more detail on each of these steps and mentioned some of the shortcomings of the Delaunay approach.
Figure 3.4. Sandstone: Portion of meshed region with elements labeled according to poremap. Volume rendering cutaway on left portion of image used to show structure of solid phase.
Figure 3.5. Meshed subdomain of random sphere pack, elements colored according to pore label (poremap). Traction boundary conditions applied at external throats (a few examples identified as red arrows) according to PN pressure solution. The subdomain extraction technique is discussed in 3.2.3.1.

3.2.3.2.1 Variable Mesh Refinement

The primary reason for choosing FEM as the streamline scale numerical technique is the ability to increase numerical resolution only where it is needed. During the point generation and distribution step (2) of the meshing procedure, additional points are added near
surfaces to ensure the mesh conforms to the geometry of the voxel image. Variable refinement allows the user to specify regions of interest and the way points are distributed.

A general surface refinement is used to improve conformance to actual pore geometry. This systematic refinement has several user-specified tuning parameters which control the distribution of additional points. The secondary resolution near surfaces specifies the refined distribution of points. Buffer size determines how far on each side of a surface to leave empty of points. This can help to minimize the number of sliver-shaped elements created during the subdivision step (4). A decay exponent is used to dictate how quickly the denser point distribution decays with distance from the surface. Finally, a maximum extent of refinement is used to limit how far from the surface the refinement is able to reach, regardless of the decay rate.

A pore network-based refinement technique was suggested by Lane (Lane 2011) and developed here to ensure tight pore throats (with radii below the average point distribution distance) would not be ‘closed’ during the meshing process. Throat refinement must be systematic as a manual refinement of the thousands to millions of throats would be non-feasible. Also, considering throat sizes in a single porous material can span orders of magnitude, one cannot simply specify a secondary resolution in all regions of throats which would be both adequately fine for the smallest throats and not computationally wasteful in the largest throats. A set of user control options similar to those of the surface refinement is available for PN-based refinement, with slight differences in how they are implemented. The refinement scheme here also utilizes throat properties which are already available in the PN, the throat locations and inscribed radii. First, the ratio of throat size to the general resolution (or surface-refinement secondary resolution, if available) is used to determine which throats are to be refined. The throat-refinement secondary resolution is determined locally by a scaling factor (>0, <1) and the inscribed radius of that throat. The range/extent of refinement is also based on a scaling factor (typically >1) and the inscribed radius. This parameter should be modified to consider the throat length as well (especially in cases of long, slender throats), but radius-based extent proved adequate in this study.
Figure 3.6. Tight pore space - 2d slice through pore-labeled meshes with (bottom) and without (top) throat refinement superimposed on poremap, demonstrating “closed” pore throat in coarse mesh which is fixed with throat-based refinement. Each color represents a different label and corresponding PN pore.

An obvious reason for using this type of refinement is to ensure all physical pathways present in the material also exist in the simulation (Figure 3.6). A comparison of pore-labeled meshes with and without PN throat-based refinement is shown in Figure 3.7 to illustrate the subtler impacts that coarsely meshed throats can have on simulation results. The two bottom images in Figure 3.7 show 2D slices through the tetrahedral meshes of a carbonate sample. The left has no refinement, while the right has PN throat-based refinement. In the labeled poremap, pores $B$ and $E$ share only one voxel face, not enough to be considered a throat under the minimum voxel restriction (discussed in 3.3.2). The refined mesh conforms well to the geometry of the void space and the elements.
labels agree with the poremap connectivity, but the unrefined mesh has several B-labeled elements sharing faces with E-labeled elements. Using this mesh for simulation will result in a significant fluid flux between \( B \) and \( E \), though all fluid transferred between them should pass through pore \( D \). This will cause highly inaccurate estimations of flowrates \( Q_{DB} \) and \( Q_{DE} \). Flowrates \( Q_{AB} \) and \( Q_{BC} \) will also be less accurate in the coarse mesh because so few interior nodes are present in throats \( AB \) and \( BC \). Having few interior nodes in a throat can also increase the likelihood of mass balance issues (See APPENDIX A).
Currently, the data structure of the PNs does not allow for multiple connections between two neighboring pores. When two pores are connected by two distinct physical pathways, there should be two identified throats – each with unique geometric characterization parameters. Currently, the properties are merged into one throat. With the throat location and inscribed radius solely corresponding to the largest of the two (or more) connections between two pores, any smaller throats will not be refined, despite being more likely to require refinement. An example is shown in Figure 3.8.
Figure 3.8. Mesh refinement when neighboring pores are connected by multiple separate throats. White and Blue pores have two connections. Since network model only identifies one maximum inscribed radius and corresponding throat location, systematic PN-based refinement will not improve meshing of the smaller horizontal throat. Notice the element density around the horizontal connection is much less than that of the vertical one. Bottom: horizontal and vertical cuts through the mesh reveal element density in each throat. The coarsely meshed horizontal throat has a single interior node shared by many solid boundary elements. This can lead to the mass balance problems discussed in APPENDIX A.

The solution to this is not very complicated, but has not yet been implemented because there are a large number of in-house network modeling codes currently in use (by the author and multiple other research group members/collaborators) which assume a single throat between any two neighboring pores. Any major changes such as this to the data structure of the PN will require a substantial effort to ensure compatibility of all codes. Other impacts of this problem are discussed further in the future work section, 5.1.
A physics-based adaptive refinement technique (as proposed by Lane) would be advantageous, but has not been developed here. One could see the benefit of adaptively refining mesh resolution in regions of very high and very low pressure drop, but should keep in mind that the external flow field will impact local flow and pressure fields, so the refinement would be specifically beneficial to the external boundary conditions which were imposed. The throat size-based refinement implemented here has the advantage of being independent of imposed flow direction, which is necessary for the sub-pore-level nature of this work. Also, when using FEM to generate reliable data for statistical learning of conductivity, all meshed throats are equally important, regardless of their significance to the macroscopic flow. A physics-based approach would be more appropriate in FEM models used directly for macroscopic property prediction.

The use of both surface and throat refinement in a hybrid modeling framework is necessary for interconnectivity agreement between FEM and PN models and to avoid the artificial tightening of pore space due to mesh coarsening (Borujeni, Lane et al. 2013).

3.2.3.3 Stokes Solver

This algorithm is an FEM solver for the Bubnov-Galerkin formulation of the Stokes, which means the weighting functions employed in the weak form are equal to the shape functions used in formulating the problem. The pore space is discretized into $P_2P_1$ Taylor-Hood elements, where the interpolation used is quadratic for velocity, and linear for pressure. There are ten nodes per element – 4 vertices used for pressure and velocity, and 6 edge midpoint nodes used only for velocity. The no-slip condition is assumed, so nodes located at void-solid interfaces are Dirichlet nodes with the velocity set to zero. Fortunately, a large fraction of the nodes in meshes of consolidated porous media consists of these surface nodes, which make no contribution to the overall matrix or to the computational effort. The $u$ component of the weak form (weak implies the differentiability of the solution has been lowered, which permits lower-order and less-smooth functions to approximate the solution) of the stress-component form of the Stokes equations is:

$$
\int_{\Omega} \left[ \left( 2\mu \frac{du}{dx} - P \right) \frac{dW}{dx} + \mu \left( \frac{du}{dy} + \frac{dv}{dx} \right) \frac{dW}{dy} + \mu \left( \frac{du}{dz} + \frac{dw}{dx} \right) \frac{dW}{dz} \right] \, d\Omega = \int_{\Gamma} \bar{\sigma}_x W \, d\Gamma \quad (25)
$$

where $\bar{\sigma}_x$ is the surface traction, which is how the boundary conditions will be applied (Lane 2011).
\[
\sigma_x = (\sigma_{xx} - P)n_x + \tau_{xy}n_y + \tau_{xz}n_z
\]  
(26)

Normal Stresses:
\[
\sigma_{xx} = 2\mu \frac{d}{dx}u - P
\]

Sheer Stresses:
\[
\tau_{xy} = \tau_{yx} = \mu \left( \frac{d}{dy}u + \frac{d}{dx}v \right)
\]
\[
\tau_{xz} = \tau_{zx} = \mu \left( \frac{d}{dz}u + \frac{d}{dx}w \right)
\]
\[
\tau_{yz} = \tau_{zy} = \mu \left( \frac{d}{dz}v + \frac{d}{dy}w \right)
\]

The \(v\) and \(w\) components are handled similarly, and the resulting three equations are converted into discrete sets of equations by employing shape (or **approximating**, or **interpolating**) functions. After discretization and replacing the weighting functions \(W\) with the shape functions \(N\) (as per the Bubnov-Galerkin formulation), equation (25) and the continuity equation (19) become

\[
\int_{\bar{\Omega}_e} \left[ \left( 2\mu \frac{d}{dx}u - N_p \right) \frac{dN_i}{dx} + \mu \left( \frac{d}{dy}u + \frac{d}{dz}v \right) \frac{dN_i}{dy} + \mu \left( \frac{d}{dz}u + \frac{d}{dx}v \right) \frac{dN_i}{dz} \right] d\Omega = \int_{\bar{r}_e} \alpha_x N_i d\Gamma
\]  
(27)

and

\[
\int_{\bar{\Omega}_e} \left[ \frac{\partial}{\partial x}u + \frac{\partial}{\partial y}v + \frac{\partial}{\partial z}w \right] N_p d\Omega = 0,
\]  
(28)

respectively for node \(i\) on element \(\Omega_e\). These can be simplified to (29) and (30) via (31).

\[
(2K_{11} + K_{22} + K_{33})[u] + K_{12}[v] + K_{13}[w] - L_1[P] = \{R_u\}
\]  
(29)

\[
L_1^T[u] + L_2^T[v] + L_3^T[w] = 0
\]  
(30)

\[
K_{11} = \int_{\bar{\Omega}_e} \mu \left( \frac{d}{dx} \right) \left[ \frac{d}{dx} \right] d\Omega; \quad K_{22} = \int_{\bar{\Omega}_e} \mu \left( \frac{d}{dy} \right) \left[ \frac{d}{dy} \right] d\Omega; \quad K_{33} = \int_{\bar{\Omega}_e} \mu \left( \frac{d}{dz} \right) \left[ \frac{d}{dz} \right] d\Omega;
\]
\[
K_{12} = \int_{\bar{\Omega}_e} \mu \left( \frac{d}{dy} \right) \left[ \frac{d}{dx} \right] d\Omega; \quad K_{13} = \int_{\bar{\Omega}_e} \mu \left( \frac{d}{dz} \right) \left[ \frac{d}{dx} \right] d\Omega;
\]
\[
K_{12} = \int_{\bar{\Omega}_e} \mu \left( \frac{d}{dy} \right) \left[ \frac{d}{dx} \right] d\Omega; \quad L_1 = -\int_{\bar{\Omega}_e} \frac{\partial N}{\partial x} N^p d\Omega; \quad R_u = \int_{\bar{r}_e} \bar{\sigma}_x \{N\} d\Gamma
\]  
(31)

Finally, (29-30) become the system of linear equations to be solved in matrix form:
\[
\begin{bmatrix}
2K_{11} + K_{22} + K_{33} & K_{12} & K_{13} & L_1 \\
K_{12} & K_{11} + 2K_{22} + K_{33} & K_{23} & L_2 \\
K_{13} & K_{23} & K_{11} + K_{22} + 2K_{33} & L_3 \\
L_1^T & L_2^T & L_3^T & 0
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
w \\
P
\end{bmatrix}
= \begin{bmatrix}
R_u \\
R_v \\
R_w \\
0
\end{bmatrix}
\] (32)

This is often given in the following notation:

\[
\begin{bmatrix}
K & L \\
L^T & 0
\end{bmatrix}
\begin{bmatrix}
u \\
P
\end{bmatrix}
= \begin{bmatrix}
R
\end{bmatrix}
\] (33)

During the assembly of the RHS, the assignment of boundary conditions is handled by assuming that a hydrostatic pressure can be applied to surfaces where Neumann planes exist. By neglecting the contribution of velocity gradients, an estimate for the three components of the traction (\(\vec{\sigma}_x, \vec{\sigma}_y, \vec{\sigma}_z\)) based on an assumed hydrostatic pressure and simplification of equation (28) is

\[
\vec{\sigma}_x \approx P \, n_x; \quad \vec{\sigma}_y \approx P \, n_y; \quad \vec{\sigma}_z \approx P \, n_z,
\] (34)

where \(n_i\) denotes the \(i\) component of the outward facing unit normal of a boundary face and \(P\) is the assumed hydrostatic pressure acting on that Neumann surface. This Galerkin formulation has been extensively validated for streamline-scale flow modeling by Lane (Lane 2011, Borujeni, Lane et al. 2013).

3.2.4 Image-Based Modeling

Image-based pore-scale modeling, or digital rock technology, is a powerful computational tool for investigating transport in porous geologic materials and gaining insight into the behavior of fluids within such materials. It has become a routine service in the Using 3D microtomography images to model fluid flow phenomena and predict properties of geologic samples has become appealing due to breakthroughs in both computer and image acquisition technologies. The early research in the field of pore-scale modeling typically involved using artificial or hypothetical porous media in attempt to draw correlations between properties which could improve understanding. Since the advent of image-based modeling, pore-scale modeling has been used as a predictive tool as well as a correlative one. The predictive capabilities of modeling a physical material make it an attractive alternative to performing expensive and time-consuming core experiments. To date, however, it has been used more so as a supplement than a replacement for experimental analysis.

The image-based approach to pore-scale modeling involves 3D image acquisition (typically by synchrotron CT or X-CT), image processing, discretization of the pore space (into a network, a grid, or a mesh), and flow simulation. Voxel data acquired from CT must go through a series of image-processing steps to identify which voxels are associated with the void space of the material. First, filtering is needed to lessen the noise in the image, where the goal is to smooth the image’s voxel intensity histogram but maintain sharp boundaries in the image, thereby assisting with the next step, image segmentation. In this step, the phases present in the data
(at minimum, two phases: solid and void) are identified and isolated. Finally, non-physical image artifacts which are brought about by image noise not removed by filtering and by errors associated with imaging hardware must be removed. Small, isolated clusters of voxels are either solid “islands” floating in space which are not physically possible, or void “holes” which do not contribute to flow through the medium (and may also be non-physical). Therefore, islands and holes should be removed prior to discretization. Once all voxels comprising the void space have been properly identified, the domain is discretized in preparation for flow simulation. The type of discretization depends on the type of flow model which will be implemented. Sections 3.2.2.1 and 3.2.3.1 describe the discretization methods used in this work (Network Model Extraction and Unstructured Mesh Generation, respectively).

For detailed reviews of imaged-based modeling, the author refers readers to excellent recent works (Blunt, Bijeljic et al. 2013, Wildenschild and Sheppard 2013, Bultreys, De Boever et al. 2016).

3.2.5 Multiscale Modeling

The traditional approach to modeling multiscale phenomena in porous media was hierarchical coupling, which assumes there is a separation between scales. Macroscale parameters are determined using experiments or micro-scale models, then passed to the larger-scale model. Pore-scale to macro-scale hybrids (discrete models coupled to continuum models) are prevalent amongst the multiscale modeling literature. In contrast, this work is considering streamline-scale to pore-scale or intermediate-scale.

The task of multiscale modeling in porous media has been approached in the past via many different methods including the multi-grid (Aarnes, Kippe et al. 2007, Durlofsky, Efendiev et al. 2007) and multiscale finite element (Hou and Wu 1997, Rank and Krause 1997, Chu, Efendiev et al. 2008, Hou and Efendiev 2009) methods, equation-free modeling (Kevrekidis, Gear et al. 2003, Kevrekidis and Samaey 2009), mortar coupling (Arbogast, Pencheva et al. 2007, Balhoff, Thomas et al. 2008), concurrent (Celia, Rajaram et al. 1993, Van den Akker 2010) and dynamic coupling (Sheng and Thompson 2013), and other coupling and upscaling techniques (White, Borja et al. 2006, Balhoff, Thompson et al. 2007, Øren, Held et al. 2009). The clear majority of these studies involved coupling of pore-scale or core-scale to continuum-scale models, whereas the proposed work involves the coupling of smaller scales.

Scheibe, Tartakovsky et al. (2007) defined hybrid multiscale numerical models as a group of methods which combine multiple models defined at fundamentally different length and time scales within the same overall spatial and temporal domain. Mehmani and Balhoff (2015) studied discrete-to-continuum hybrid models for fluid flow and solute transport. The work of Sholokhova, Kim et al. (2009) mentioned in 3.2.2.3.3 is another hybrid modeling approach, specifically network flow modeling via lattice-Boltzmann based channel conductance.
The Heterogeneous Multiscale Method (HMM) is a general framework for linking models at different scales rather than a particular type of model. It is a data-based coupling strategy, which makes it more flexible than solution-based counterparts. The structure typically consists of an incomplete macroscale model with the microscale model used as the supplement for missing information, where the macro state provides the environment (boundary conditions) for the micro solver. Weinan et al. provide an excellent introduction to HMM (Weinan, Engquist et al. 2007, Yue and Weinan 2007). The authors contrasted the objectives of modern multiscale techniques from traditional ones which are typically micro-solvers, with multi-grids or adaptive refinement, and therefore the computational cost is nearly that of full microscale simulation. New multiscale methods aim for a cost far lower than exhaustive microscale simulation. They also split multiscale problems into four categories:

A. Problems containing isolated defects (cracks, dislocations, contact lines, etc.) where the microscopic model is only needed in the region of the defect.

B. Problems where a closed macroscopic model is needed for a properly selected set of macroscopic variables, but it is not explicit enough to be used directly.

C. Problems with a mixture of the features of A and B problems

D. Problems with self-similarity in scales. As opposed to Type A and B problems, which benefit from scale separation.

The framework proposed in this work is a type-C HMM with some features of hybrid models, where the macroscale model will be the pore-scale network model and the microscale model will be the streamline-scale FEM.

Another type of multiscale modeling at the pore-level involves dealing with the multiple spatial scales inherent to the physical pore space of a porous material. Often, a single imaging experiment cannot capture all the pores of a material due to limited resolution and sample size (Bultreys, De Boever et al. 2016). This can be overcome by a dual-porosity approach, where multiple images are used for each material. Dual pore network models distinguish between micropores and macropores. The handling of these dual-porosity systems has been approached in numerous ways, including combining all pores (micro and macro) into a single PN (Jiang, van Dijke et al. 2013), treating the microporosity between macropores either as a continuum media or as a new type of connection called “micro-links” (Bultreys, Van Hoorebeke et al. 2015). This is an ongoing area of research with much interest.

3.2.6 Improving Network Parameters with Machine Learning

This study proposes a pore-level, multiscale, data-driven framework for porous media flow modeling that will utilize modern techniques and technologies, including as x-ray computerized tomography, high performance computing, finite element method, pore network modeling, and machine learning algorithms. Recently (nearly in parallel with this study), van der Linden, Narsilio et al.
(2016) also built a framework using these technologies – The most similar work I’ve seen to the current project. The fact that such a similar, though completely independent, study was published so recently affirms that this project is timely and valuable to the porous media research community and the larger geological/petroleum/civil/chemical engineering communities. Like the current work, they used network modeling, FEM, and machine learning algorithms in an image-based setting to gain insights into how the microstructure of the pore space impacts macroscale flow phenomena of porous media. The specific objectives of our work differ in that we are using the machine learning to improve the pore-scale model so that it can then be used to predict macroscale parameters (including permeability, particle retention, etc.) instead of focusing on permeability directly. In other words, we characterized throat conductivities (pore-scale parameter), while van der Linden et al. characterized permeability (continuum-scale parameter). They reported that a link exists between high permeabilities and efficient shortest paths that thread through relatively large pore bodies connected by high conductance pore throats, so their predictors were based on connectivity and pore structure. Because they were concerned with permeability, many images were required to generate a sufficient amount of training data for machine learning – more than could feasibly be obtained through CT imaging alone. Discrete element method was used to generate sphere packs, and an CT image of glass beads was also used.

The term *machine learning* has become popularized in recent years thanks to advances in computer architecture and software, data collection and distribution, and exciting research in the field of AI. The concepts of predictive linear modeling have been in use since the early 1800s (Gauss and/or Legendre credited with least squares around that time), and very complex and robust statistical techniques have since been developed and used continuously by mathematicians and statisticians. In the last couple of decades, statistical predictive modeling has seen new popularity amongst a broader community of engineers, scientists, businesspeople, and even hobbyists because of the accessibility to open source libraries, inexpensive computational resources, and an abundance of data. The pore-scale modeling research community also has access to these assets, so it is fitting (pun intended) that these popular techniques should be experimented with and utilized. Only regression algorithms will be discussed in this work, though classification and unsupervised methods are also remarkably powerful tools.

Modern machine learning techniques can often offer greater predictive accuracy than traditional statistical approaches, but it is often at the expense of interpretability. Machine learning is also extremely useful as a tool for exploratory data analysis, to find the important predictors, detect outliers, and gauge what kind of "smoothness" your data has. One can create a large feature space and use statistical algorithms to identify which features are important and which are unimportant, or worse – harmful to the model due to correlated noise/outliers.
In the PN-FEM hybrid model, the features/predictors for hydraulic conductivity are obtained strictly using information from the PN, so that future predictions can be made using only the information available to the PN. Basic geometric and connectivity-based features are taken directly from the PN, and other more intelligently chosen features are computed. A full list of used features is provided in 3.3.5.

To use machine learning in pore-scale modeling, we must generate a data set where known hydraulic conductivities are the target (dependent) variables, and the feature set is composed of the measured properties of the pores and throats. The known conductivities are determined by sub-pore-scale modeling, specifically FEM. Physically-representative pore networks already compute the geometric properties of the pores and throats, so extracting a feature set is a matter of determining the corresponding PN parameters for the FEM-computed throat conductivities.

Once a data set is generated, the examples are split into a training set (used for training the learning model) and testing/cross-validation sets. Feature selection and regression algorithms are then used to learn an appropriate model of the training data and the test data is used to analyze the model’s performance (and how well it generalizes).

It is recommended to use stratification when splitting up the data. The training, cross-validation, and test sets should all have statistically similar distributions of data. Simply using a random split can result in most of the outliers ending up in only one of the sets. This would cause either a badly trained model or an erroneously high reported accuracy. In either case, the results would not be useful for future predictions.

3.2.6.1 Regularization, Feature Selection, Model Simplification

In machine learning (and numerical modeling in general), it is important to keep in mind that the most complex models are rarely the best models. When dealing with feature spaces with many dimensions, this is true for many reasons. Some reasons for simplifying a model are:

1. **Avoiding Overfitting and High-magnitude Coefficients**: Most important reason for the current project. Since we want to use the model to make accurate predictions on data outside the training set, the bias-variance trade-off is very important.

2. **Parsimony and Interpretability**: A parsimonious model is the simplest model with the least assumptions and variables, but with greatest explanatory power. Machine learning algorithms are notoriously described as “black boxes”, meaning they tend to be unfavorable for situations where it is desired to gain some insight into the relationships between the parameters and the outcomes, rather than simply producing accurate predictions. Simplifying a model can have the added benefit of shedding light onto these relationships.
3. Minimizing Computational cost: This can be a factor since very complicated models tend to be computationally expensive, though this reason has not influenced the model selection in this work for a few reasons: we are fortunate to have a lot of computational power at our disposal; the training data is generated by the computationally expensive FEM model, where creating meshes is the bottleneck in the overall workflow. Therefore, training set sizes are typically on the order where even very complicated models run relatively quickly; and most importantly, I am not an expert in machine learning and therefore decided to stick to simpler models initially.

Having too much information is a good problem to have, compared to the alternative. There are many ways to go about whittling down the complexity such as choosing simpler algorithms, feature selection, and regularization.

3.2.6.2 Algorithm Selection

There are many options of machine learning algorithms for predicting continuous variables, including linear regression, polynomial regression, decision (regression) tree, random forest, support vector regression, and artificial neural networks. Each has strengths and weaknesses, and (unfortunately) there is no “one size fits all” algorithm which is guaranteed to give good results for every problem. This is an over-simplification of the infamous “No Free Lunch” theorem (Wolpert 1996), which basically says that no learner can perform better than random guessing over all possible functions to be learned. Known relationships between predictor and target variables can help narrow down the list of options, but choosing an algorithm often comes down to tradeoffs between speed, available computational resources, required accuracy, model interpretability, user expertise, and (possibly most importantly) amount of training data available. I am fortunate to have ample computer resources at my disposal, so speed and resources are not factors for the data sets of interest here (we are not yet in the realm of Big Data). The importance of interpretability is debatable. For the purposes of a dissertation, however, it is preferable to understand why an algorithm makes a particular prediction rather than taking the true black box approach (artificial neural networks, etc.). For this reason, there will be an emphasis on linear regression models and regression trees, rather than k-Nearest Neighbors (and Kernel Regression) and Artificial Neural Networks, for this study.

3.2.6.2.1 Parametric and Nonparametric Learning Algorithms

Parametric methods involve fitting parameters to an assumed model, as opposed to the nonparametric approach of determining the model from the data. Typically, parametric algorithms are simpler, faster, and require less data, while nonparametric algorithms are more flexible, powerful and accurate. Ordinary Least Squares (OLS) regression is the most popular parametric method of fitting a
linear model to training data and Gauss-Markov Theorem states that it is the best linear unbiased estimator. In OLS, the coefficients \( \theta \) are computed such that the residual sum of squares (RSS) is minimized.

\[
RSS(\theta) = \sum_{i=1}^{m} (y_i - x_i^T \theta)^2
\]  

(35)

In matrix form, with \( X \) being an \( m \times n \) matrix where each row is a training example and each column is a feature/predictor,

\[
RSS(\theta) = (y - X\theta)^T (y - X\theta).
\]  

(36)

The minimization can be done using the gradient descent method, or if \( X^T X \) is nonsingular, the unique solution to the normal equation is

\[
\hat{\theta} = (X^T X)^{-1} X^T y.
\]  

(37)

K-Nearest Neighbors (KNN), Kernel Regression, and Decision/Regression Trees are a few popular nonparametric algorithms, in which complexity can grow with number of data points. Bias can theoretically approach zero if there is no noise and infinite data, as opposed to parametric models which will have inherent model bias that cannot be overcome by cleaning or additional data points (e.g. fitting a sinusoid with a quadratic function – it doesn’t matter how many data points fall on the sinusoid, there will always be some nonzero bias). Nonparametric algorithms suffer when available training data is limited relative to the dimensionality of the input space. So as the number of features increases, the amount of data necessary to use a nonparametric approach increases exponentially. Decision and regression trees use stratification of the predictor space to create splitting rules which can be used to make predictions in a manner similar to human decision making (James, Witten et al. 2013). The predictor space is divided into many distinct regions. The prediction for observations which fall into a particular region of the feature space is simply the mean of the response values of the training data in that region. While tree-based methods are simple and among the most interpretable models, they are not known to have high prediction accuracy compared to advanced linear and nonlinear regression algorithms. They can outperform classical linear models when there is a complex, non-linear relationship between the features and response. Random Forest is an ensemble method (many weak learners vote on the best solution) where multiple decision/regression trees are combined to generate a consensus prediction. They are random in that a random small subset of features (e.g. ~ the square root of the total number of features) is used for training the decision trees and only one feature is considered at each split. Regression trees can suffer from being non-robust, but the trees in the forest are intentionally decorrelated from one another, so that the average of the tree predictions is less variable. This has been shown to have greater prediction accuracy than a single boosted tree, though interpretability is significantly lessened (James, Witten et al. 2013). Boosting is a way to improve regression trees by building a
sequence of trees which focus on the scaled residuals of the previous tree. This is done to increase (“boost”) the importance of badly predicted observations, yielding markedly better prediction performance.

3.2.6.3 Bias variance tradeoff

In the current work, we are ultimately concerned with test, or prediction, error more so than the training error. This is the error in predicting values for data which was not used to train the model. This type of error can be caused by either bias or variance. Models suffering from high bias underfit the data, whereas high variance results in overfitting. Using all available data to train a model, then reporting the prediction error on the same data is a bad approach because the reported accuracy of the model will be inflated (since the model was specifically tuned to minimize the error of that data specifically), and it is not likely to generalize well to accurately predict other (even very similar) data due to overfitting (which defeats the entire purpose of creating a model in the first place). Cross-validation techniques are used to tune models in such a way that the test set error can be minimized, rather than the training set error. The model with the best bias-variance tradeoff is the one with the minimum test set error between many models of increasing complexity. Examples of models suffering from high bias and variance are illustrated in Figure 3.9.

![Figure 3.9. Example of models of varying complexity. Blue data represent the training set, Orange data represent the test set. Left: model suffering from high bias. Right: model suffering from high variance. Center: model with low bias and variance.](image)

The bias and variance also depend on the size of the data set available for learning from. The amount of training data available is not a stagnant number, since it is generated with numerical simulation and more images can be obtained. For a sample of interest, however, we are limited by the number of throats in the image (which depends on the type of rock and image size/resolution, and can be on the order of 100,000s to millions) and the amount of time that is required to produce target variables in the training data, which are obtained through expensive direct numerical simulation. Considering we are looking to improve network modeling capabilities, and not to create the best possible machine learning model, it is counterproductive to run DNS on many large images only to use the information in a less accurate model. We hope to take advantage of the computational efficiency of the network, so learning curves (Figure 3.10) will be used to determine the smallest amount of training data. When training data is limited,
subdividing it in the traditional holdout validation procedure (e.g., 25% of the data is only used for cross-validation, 25% is only used for testing, and the remaining 50% is used to actually train) can lead to insufficient set sizes and models with high variance. Applying techniques such as K-fold cross-validation allows one to perform model validation without “wasting” a large portion of the training data.

Figure 3.10. Anticipated shape of learning curves, which are used to determine adequacy of training set size. Actual “true” error would assume the data is noiseless. Since this is not a valid assumption, test set error will indicate model performance and generalizability.

3.2.6.4 Feature Engineering and Feature Selection

Producing highly-predictive features from the available data using domain knowledge is vital to the success of predictive models. Purely mathematical procedures for feature engineering (polynomial combinations, etc.) and extraction (Principal component analysis to lower dimensionality) reveal only a limited amount of additional information on their own, unless an extreme amount of combinations is tested. Using domain knowledge both improves performance in a more efficient way, and has the added benefit of being more interpretable. In this study, the target/independent variable has a physical meaning: the linear relationship of the pressure drop across a throat to the flowrate through it. So, knowing that the relationship between flow and pressure drop in a tube follows the Hagen Poiseuille equation, it makes sense to use $R_{ij}/L_{ij}$ as a feature in predicting hydraulic conductance. Since the conception of network models, researchers and engineers have developed many hydraulic conductivity models of varying complexity to relate pressure to flow in throats. This problem has been approached experimentally, theoretically, and statistically (see section 3.2.2.3 on $g_{ij}$ methods). Most models have been specific to a material, or a family of throat shapes while others are more general. By including several of these traditional models as features of a linear model, we are performing weighted model...
averaging. By doing so we make no broad assumptions about the uniformity of throat geometries in the material, or which single model is best for a given material.

The first feature selection approach one should take is also utilization of domain knowledge. In other words, thinking critically about the type of model being used, whether each feature can have an impact (positive or negative) on the accuracy of the prediction, and whether any features are likely to be highly correlated.

Once features have been suggested, algorithmic strategies for feature selection can be employed. Feature selection methods fall into three classes: Filter, Wrapper, and Embedded methods (Guyon and Elisseeff 2003). Filter methods select a subset of variables as a preprocessing step and tend to be faster than wrapper methods. They are often in the form of univariate tests which score the performance of each feature in relation to the target, eliminating the poor performers. For this reason, they lack any ability to address multicollinearity or identify correlations amongst features, so all features must also be tested pairwise to address this issue. Furthermore, filter methods can fail to find the best subset of features since feature interactions are not accounted for and two variables that are useless by themselves can be useful together (Guyon and Elisseeff 2003). Examples of filter-type tests include Pearson’s Correlation, Linear Discriminant Analysis, Analysis of Variance, and Chi-Square.

Wrapper methods, can find the best subset of features typically performed iteratively (making them rather computationally expensive), usually on the same algorithm that is used to train the final model. Forward selection starts with only the intercept (the model simply predicts a constant – the mean of the training data) and sequentially adds predictors to the model, while backward selection starts with the full model and sequentially deletes predictors that have the least contribution to model fit.

Embedded methods perform both continuous shrinkage and automatic variable selection simultaneously. The lasso method (Tibshirani 1996) is a form of $L_1$ regularization, where OLS is modified so that the absolute sum of coefficients is minimized. Similarly, ridge regression (Hoerl and Kennard 1970) minimizes the sum of the squared coefficients ($L_2$ regularization). In both cases, a penalty term is added to the ordinary least squares regression algorithm in order to achieve two quite valuable benefits: to shrink coefficient magnitudes and to result in models with lower variance (less overfitting). The lasso has the added benefit (over ridge, assuming that sparsity is advantageous) that it tends to result in sparse models. That is to say: the ridge regression results in predictor coefficients which are lower than OLS coefficients, but nonzero; whereas the lasso tends to produce 0-valued coefficients, thus eliminating predictors from the model. Lasso is popular because it simultaneously produces accurate and parsimonious models (Zou, Hastie et al. 2007) with fewer degrees of freedom than OLS or ridge. Ridge regression would be preferable in cases where all predictors are important, since it always keeps all the predictors in the model (Zou and Hastie 2005). In cases where multiple features are highly
correlated, the lasso will effectively pick one of them to keep at random; the ridge would keep them all. Neither approach is ideal, and for this reason the elastic net regularization was developed. Elastic net is a hybrid of lasso and ridge techniques, which often outperforms lasso in terms of prediction accuracy (Zou and Hastie 2005). Like lasso and ridge, it is a penalized least squares method. Whereas the lasso penalty is indifferent to the choice among a set of strong but correlated features and the ridge penalty shrinks the coefficients of such features toward each other, the elastic net penalty is a compromise. The parameter \( \alpha \) can be varied to encourage highly correlated features to be averaged or to encourage sparsity in the coefficients of the correlated features. The penalty terms for these three regularization techniques are the last terms in the following equations. Each of these comes from the very informative Elements of Statistical Learning (Hastie, Tibshirani et al. 2009).

Ridge:

\[
\hat{\theta}_{\text{Ridge}} = \arg \min_{\theta} \sum_{i=1}^{m} \left( y_i - \theta_0 - \sum_{j=1}^{n} x_{ij} \theta_j \right)^2 \text{ subject to } \sum_{j=1}^{n} \theta_j^2 \leq t
\]

\[
\hat{\theta}_{\text{Ridge}} = \arg \min_{\theta} \left\{ \sum_{i=1}^{m} \left( y_i - \theta_0 - \sum_{j=1}^{n} x_{ij} \theta_j \right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right\}
\]

\[
\hat{\theta}_{\text{Ridge}} = (X^TX + \lambda I)^{-1}X^Ty
\]

Lasso:

\[
\hat{\theta}_{\text{Lasso}} = \arg \min_{\theta} \sum_{i=1}^{m} \left( y_i - \theta_0 - \sum_{j=1}^{n} x_{ij} \theta_j \right)^2 \text{ subject to } \sum_{j=1}^{n} |\theta_j| \leq t
\]

\[
\hat{\theta}_{\text{Lasso}} = \arg \min_{\theta} \left\{ \sum_{i=1}^{m} \left( y_i - \theta_0 - \sum_{j=1}^{n} x_{ij} \theta_j \right)^2 + \lambda \sum_{j=1}^{n} |\theta_j| \right\}
\]

Elastic Net:

\[
\hat{\theta}_{\text{Elastic Net}} = \arg \min_{\theta} \sum_{i=1}^{m} \left( y_i - \theta_0 - \sum_{j=1}^{n} x_{ij} \theta_j \right)^2 \text{ subject to } \sum_{j=1}^{n} (\alpha \theta_j^2 + (1 - \alpha) |\theta_j|) \leq t
\]

\[
\hat{\theta}_{\text{Elastic Net}} = \arg \min_{\theta} \left\{ \sum_{i=1}^{m} \left( y_i - \theta_0 - \sum_{j=1}^{n} x_{ij} \theta_j \right)^2 + \lambda \sum_{j=1}^{n} (\alpha \theta_j^2 + (1 - \alpha) |\theta_j|) \right\}
\]

To overcome the double shrinkage of the quadratic penalty, which introduces unnecessary bias, a scaling correction \((1+\lambda)^2\) is introduced. This has the added benefits of improving the prediction performance over ridge regression and being more stable than the lasso (Zou and Hastie 2005).
\[ \hat{\theta}_{\text{Elastic Net}} = \arg \min_{\theta} \theta^T \left( \frac{X^TX + \lambda_1 I}{1 + \lambda_2} \right) \theta - 2y^TX\theta + \lambda_1 |\theta|_1 \]  

(42)

3.2.6.5 Data Cleaning and Preparation

Linear regression models make several highly restrictive assumptions which are rarely upheld by raw data sets in practice (James, Witten et al. 2013). Common assumptions which can be problematic:

1. Linearity – linear relationship between response and predictor variables
2. Homoscedasticity and uncorrelated error terms – error terms have a constant variance and follow no trends
3. Normal distribution – data follows a Gaussian distribution (or nearly so)
4. Low Noise – no flagrant outliers or very noisy data
5. No collinearity (or multicollinearity) – each predictor is linearly independent

Linear models can often be extended to data sets which violate these basic assumptions using the following techniques which relax these assumptions.

3.2.6.5.1 Transformations

Non-linearity can be addressed by using polynomial regression. In many cases linear regression can still be successfully implemented on nonlinear data through the use of data transformations. Common transformations used on the predictor variables to address non-linearity include \( \log(X) \), \( X^{1/2} \), and \( X^2 \).

Another reason for using data transformations is to address heteroscedasticity, or the presence of non-constant variance of error terms. By using a concave-down function for transformation of the response variable (such as \( \log(Y) \) or \( Y^{1/2} \)), the right tail of the data distribution is pulled in and the left tail is pushed out. Box-Cox is another such transformation. These transformations were tested but found to have insignificant effect, sometimes improving prediction performance, but more often increasing error.

3.2.6.5.2 Noise Removal

Removing erroneous data points is especially important in the output variable as SSE means errors will be squared and extreme outliers will cause skewed models. Visual inspection of the distribution of features can aid in the removal of noise.

In the context of this work, most noise comes from the presence of network and/or mesh anomalies, which cause strange, non-physical flow scenarios, resulting in unexpected values of hydraulic conductance. For instance, throats where:
• Wetted perimeter is 0 or a very small fraction of total perimeter – a throat is defined as a constriction in the pore space, so the wetted and total perimeters are equal in ideal cases. This is often not the case because of irregular 3D geometries of real materials and pore merging during network extraction. The right side of Figure 3.11 gives an example of cross-section through a throat where the total perimeter is not in contact with the solid-void interface. The non-wetted portion (shown in yellow) is relatively low compared to many throats in PNs extracted from real geologic materials.

![Figure 3.11. Throat perimeter examples, viewing throat cross-sections. Wetted perimeter shown in green, non-wetted portion in yellow. Left: Entire perimeter is wetted. Right: a portion of the perimeter is non-wetted.](image)

• Throat length is very small, leading to extremely small pressure drops between neighboring pores. This noise was lessened by filtering out data points (throats) where the pressure drop was below some threshold. By very small, we mean only a few voxels between pore pressure interpolation locations.

• As mentioned in 3.2.2.1, the parameters used for the PN extraction process can impact the quality of the data available to train a regression model. Higher pore density yields a larger number of training examples per domain, but they are less representative of typical pore throats. While learning algorithms benefit from abundance of data, the features available for learning are based on physical pore throat parameters. These assumptions fall apart when many pores are overlapping. At the same time, low pore density makes agreement between PN and FEM models difficult because of large distributions of pressures within single pores (discussed further in 3.3.4.1). The maximum number of pore merges is a control parameter in the PN extraction algorithm. It must be adjusted to balance the two problems.

• Mass balance issue (discussed in APPENDIX B) – The Galerkin FEM formulation is overall mass conservative, but not explicitly conservative element-wise or pore-wise. Small pores and throats modeled by only a few elements tend to have
significant mass imbalance pore-wise. Using these flowrates to compute hydraulic conductivity produces inherent noise since it is a departure from the physical law of mass conservation which the PN model is based on. Therefore, only throats where relative error in flow balance was below some cutoff were kept in the training data set (in this work 25% was used to increase the size of the data set available for modeling, but stricter tolerances would be preferable.) A mass conservative FEM formulation may reduce noise but at the costs of lower accuracy and more costly computation in the streamline-scale model.

- Coordination number is defined as the number of pores a pore is connected to in the PN. If either pore has a coordination number of 1, indicating it is connected to only one other pore, then it is a dead-end pore and the true net flow through the throat is 0. There may in fact be a nonzero pressure drop between pore centers, and the circulation pattern of the streamline-scale velocity solution may reveal flow into and out of the dead-end pore. This typically results in radically unrealistic conductivities, and often in negative ones, regardless of the other properties of the throat (predictors/features). Therefore, while coordination number may not have much predictive power in regression models, it is quite useful to use in the data cleaning process. It should be noted that not all dead-end pores are identified by their coordination number. Like a single dead-end pore, a cluster of pores can also be stagnant, yielding noise which is not so easily removed.

Some of the scenarios mentioned above were found using 3D exploratory visualization, while others were identified using a combination of feature distribution plots, scatterplot matrices, and KNN or Mahalanobis Distances for multivariate outlier analysis in the JMP statistical software package. Of course, there will always be some degree of noise which can't be accounted for since there are a limited number of geometric parameters, and they do not describe pore—throat geometries uniquely. This is especially true in real materials.

3.2.6.5.3 Feature Rescaling

Standardizing or normalizing input variables will often improve the reliability of linear regression predictions, and will result in far faster convergence for gradient descent, though some regression algorithms are scale-invariant. A common approach to feature scaling is to get all feature values to be centered at 0 and have the same order of magnitude using mean-normalization. This can be done by subtracting the mean value (for each feature) from each and dividing by either the range or the standard deviation:

\[ x'_i = \frac{x_i - \mu_i}{s_i} \quad s_i = \text{range or std for feature } i \]  \hspace{1cm} (43)

One should keep in mind that when a model is trained on scaled features, future predictions must be made on features on the same scale. This means that the means, ranges, and standard deviations should be stored along with any learned expression.
3.2.6.6 No Free Lunch, P-Hacking, Data Leakage

While it is important to consider multiple approaches to solving a given problem, there is another trade-off (besides the bias-variance trade-off) which is subtler, but should still be taken into consideration when extensively analyzing large amounts of data. The No Free Lunch Theorem, mentioned previously, suggests one should not just take an off-the-shelf algorithm and run with it, assuming it will output the best model for any given data set. Still, it is ill-advised to train an extreme number of slightly different models with varying algorithms and hyper parameters and then just choose the one which has the lowest test set error. This approach would likely yield a model which has overfit to the noise in the test set and would not generalize well to new data. The term p-hacking is used to signify that one has just brute-force tweaked various model inputs until a desired model probability was reached, rather than taking a more scientific approach to model tuning. This is exacerbated by the fact that almost all algorithms have adjustable (hyper) parameters and model selection is necessary to obtain the best generalization. We are therefore required to split the available training data in to multiple sets (training, cross-validation, test), drastically lessening the amount of training examples available for model training.

Related to p-hacking, the term data leakage refers to situations where unexpected information makes its way into the training data or model parameters, specifically information that will not be available in future applications of the model for prediction. This info often comes from the test set or from the data acquisition process. Therefore, models trained with leaked information make unrealistically good predictions on the designated test set. An example of data leakage is given at the end of the explanation of the Iterative Network Pressure (INP) algorithm.

3.3 Methodology

While FEM was used in this work for the reasons described in 3.2.3, the workflow outlined here can easily accommodate other streamline scale models such as finite volume or LBM. It is worth mentioning this at the start, since many network modeling research groups use LBM and there have been recent advances in unstructured LBM (Misztal, Hernandez-Garcia et al. 2015) that can overcome some of the drawbacks that made LBM unfavorable for this work. Initially, a goal of this study was to couple the two models (network and FEM) in order to take advantage of the strengths of each. It was found that a sensible way to do this is through boundary conditions and the hydraulic conductance parameter, $g_{ij}$, which is where a lot of the assumptions in the network model are made. Since there are no adjustable parameters in the Stokes FEM, transferring information from the network to the FEM must be done through boundary conditions and/or fluid property changes. This doesn’t make the iterative approach that was first proposed any less useful. Realizing this did, however, lead to added focus on the more simplified sequential/hierarchical coupling. In other words, I realized that it was useful to just pass the $g_{ij}$ info from the FEM to the NW when possible, instead of having the
models continuously communicating back-and-forth. In the context of predicting hydraulic conductances, there was little noticeable advantage to sending information back to the FEM after running the “improved” network model.

Figure 3.12. Hybrid Modeling Workflow.

The following steps are an overview of the workflow for the multiscale framework which was used. These are also reflected in Figure 3.12.

1.) Acquire or generate 3D voxel image; Process and segment image.
2.) Generate the pore network model (using Vox2Net algorithm), labeling voxels with network pore indices.
3.) Identify a subdomain (or multiple subdomains) of the image in which to perform FEM simulations.
4.) Within the subdomain, generate the FEM mesh from the labeled voxel image using in-house meshing codes with PN throat-based refinement.
5.) Reconcile mesh and network pore connectivities.
6.) Run single-phase network flow model on the PN using a default throat hydraulic conductance formula.
7.) Use the PN pressure solution to determine boundary condition values which are applied at the faces of throats on the subdomain boundary.
   a. Throat pressures are computed using radius- or distance-weighted averaging of neighboring pores to determine throat pressure values.
8.) Run single-phase stokes FEM code to compute pressure and velocity fields within the meshed region.
9.) Compute throat hydraulic conductivities ($g_{ij}$). Compute throat flowrates by integrating the velocity across all throat areas.

Use network pressures or interpolate FEM pressure fields to get pressure drops across each throat.

10.) Use machine learning algorithms to develop a model for future $g_{ij}$ prediction.

11.) Use learned model to predict $g_{ij}$ in every throat of the network model, recompute the network solution using these learned values.

The following sections give additional details on the steps of the workflow which haven’t already been thoroughly discussed in previous sections.

3.3.1 Image Acquisition

The CT images used in this work were acquired from the Digital Rocks Portal (Prodanovic, Esteva et al. 2015), which is an open data repository for sharing and scientific cross-validation of volumetric images and analyses of porous materials. A sintered glass bead pack was created and imaged by Khan et al (Khan, Prodanovic et al. 2016) to obtain a sample with high porosity (41.6%) and uniform grain size (~1mm). It was imaged at a resolution of (10 μm)$^3$/voxel, using x-ray microtomography and used to study particulate straining (Khan, Mirabolghasemi et al. 2016). Muljadi et al (Muljadi, Blunt et al. 2016) studied the impact of pore-scale heterogeneity on non-Darcy flow behavior using images of Bentheimer sandstone and Estaillades carbonate, imaged at (3.0035 μm)$^3$/voxel and (3.3113 μm)$^3$/voxel resolutions, respectively (Muljadi 2015, Muljadi 2015). 2d slices of these images are shown in Figure 3.14.

The glass bead pack data was provided as a stack of greyscale slices which required cropping and segmentation prior to model discretization (Figure 3.13). Segmentation was carried out using an in-house Indicator Kriging code developed by Bhattad (Bhattad, Willson et al. 2010), based on the work of Oh and Lindquist (Oh and Lindquist 1999). The final processed image was 1200 voxels$^3$, providing a (12mm)$^3$ domain for pore scale modeling. The sandstone and carbonate were segmented prior to distribution, and had domain sizes 1000$^3$ voxels and 650$^3$ voxels, respectively (approximately 3.0mm and 2.2mm).

An isotropic, polydisperse, random sphere pack was generated in-house and was also used throughout the study (Figure 3.3).
Figure 3.13. Slice through Sintered Glass Beads CT data before (left) and after (right) cropping and image segmentation. The greyscale image (left) appears to be already segmented because it was extremely clean, making noise-filtering unnecessary and segmentation straightforward.

Figure 3.14. Slice through segmented Benheimer Sandstone (left) and Estaillades Carbonate (right) CT data

3.3.2 Connectivity Matching

After creating both the PN and FEM mesh discretizations of the pore space, one must ensure the same pore-to-pore connections exist in both (network and FEM) models. This is necessary to maintain indexing and make appropriate comparisons between them.

Thompson, Willson et al. (2008) discussed the emergence of non-physical throat connections due to a combination of voxel burn connectivity and image resolution. These present themselves as small connections between corners of pores. Typically, these
connections are less than a few voxels in diameter, and can therefore be eliminated by enforcing a minimum voxel restriction on throats. Image resolution and material of interest will dictate whether this is a well-advised strategy. If the image resolution (cm/voxel) is not several times smaller than the minimum expected physical throat, imposing a minimum voxel restriction will eliminate physical throats as well as non-physical ones. The authors recommended small or no minimum voxel cutoffs when modeling was the primary focus since the extra small throats will have negligible effect on most transport processes. In situations where network statistics were important (especially coordination number and average throat properties), they recommended analyzing the distribution of pore and throat sizes and setting the minimum on a case-by-case basis.

Figure 3.15. The two possible disagreements are 1.) network throats don’t have corresponding mesh connections (throat between left and middle pore), and 2.) mesh connections do not have corresponding network throats (no throat between right and middle pore)

Using the meshing scheme discussed in 3.2.3.2 with network-based refinement, the likelihood of throats being “closed” in the mesh can be substantially lessened. The opposite problem is also possible – where connections between pores exist in the mesh but corresponding throats do not appear in the network. This happens when network throats are removed due to a minimum voxel/throat restriction or mesh resolution was (locally) insufficient to maintain conformance to the structure of the poremap in the region of the throat. Figure 3.15 gives 2d illustrations of both. One must carefully select meshing and network extraction parameters to reduce the occurrence of these problems, with the ultimate goal being accurate physical representation of the physical pore space. Still, discrepancies in interconnectivity will often exist between the two models, so another step in the workflow is to enforce agreement in the models’ interconnectivity. This can be done by converting certain nodes in the mesh into Dirichlet no-slip
boundaries, thus damming the flow through a meshed throat. Additional throats can also be removed from the network (simple to do this) or added to the network (the new throat’s properties would not be available unless recomputed from the image – a nontrivial task while maintaining the properties of other throats and consistency with the poremap).

3.3.3 PN-Boundary Conditions and the FEM Stokes Solver

Once the two models are in agreement, the PN simulation is performed to get the initial distribution of pore pressures. The traction boundary conditions to the FEM subdomain are applied based on the PN pressures at the throats bounding the subdomain. This makes the flow field more physically-realistic than applying arbitrary directional pressure drop (along a Cartesian plane).

A significant portion of the time and computational effort involved in finite element modeling is dedicated to the setting up of the stiffness and sparse storage matrices. Fortunately, it is not necessary to complete these steps in every iteration when working to match the fluxes between the FEM and NM solvers. In each optimizing iteration, the throat hydraulic conductance values ($g_{ij}$ for throat connecting pore $i$ to pore $j$) are the parameters which are adjusted. The entire pressure solution is recomputed via the NM with new $g_{ij}$ values, then the FEM solver RHS (right-hand side) is updated with the new throat pressure Neumann boundary conditions. There is no need to recompute the stiffness matrix. Furthermore, using a matrix decomposition method, the matrix decomposition (the most computationally expensive part of solving the linear system of equations) needs only be performed one time. The decomposition matrices can be stored and reused to solve for each new RHS. This is only possible because of the use of network boundary conditions: since all external boundary conditions are Neumann-type, where traction is applied based on the network solution of pressures in neighboring pores. Previously, Neumann BCs would be applied to two sides of the FEM mesh (inlet and outlet), and no-slip Dirichlet BCs would be applied to the other four sides of the FEM domain. These arbitrary BCs are now enforced on the larger PN, which encompasses the FEM subdomain. Therefore, the FEM RHS is changing in each iteration, but the stiffness matrix is fixed for each mesh.

Pardiso is a parallel direct solver of Intel’s Math Kernel Library which calculates the solution of a set of sparse linear equations with multiple right-hand sides. The extremely sparse structure of FEM systems encountered in porous media modeling makes it a good choice for the problems of interest. It has functionality allowing users to call different phases of the solver’s execution, and the ability to keep the decomposition matrices in internal memory. Pardiso only needs to be called in the first simulation of the first iteration to decompose the matrix using numerical Cholesky factorization. Then, during successive iterations and/or imposed flow directions, the algorithm only assembles new RHS arrays to be sent into Pardiso for solving and iterative refinement.
3.3.4 Hydraulic Conductance Computation

The FEM-computed hydraulic conductivity of each throat in the meshed subdomain is determined by rearranging equation (20) and using the FEM solution to determine the parameters on the right-hand side. A viscosity equal to unity is used in the FEM simulations, so the conductivity of the throat connecting pores $i$ and $j$ is

$$g_{ij} = \frac{q_{ij}^{\text{FEM}}}{(p_j^{\text{FEM}} - p_i^{\text{FEM}})}.$$  \hspace{1cm} (44)

The FEM throat flowrates require a more rigorous computation. A surface integral of the velocity field, $u$, must be computed over the throat area. For the same throat,

$$q_{ij}^{\text{FEM}} = \int_{A_{ij}} u \cdot n \, dA_{ij}. \hspace{1cm} (45)$$

Gaussian Quadrature is used for numerical integration over the throat's cross-sectional area, $A_{ij}$. This is done on each of the element faces which make up the throat interface. Examples are provided in APPENDIX B. The throat surface is identified as the collection of faces shared by an $i$-labeled element and a $j$-labeled element.

The FEM pressure field is used to determine a single pressure for each PN pore. This is a complicated matter, discussed further in the following section. Basically, the element containing the pore location is used for interpolation of the pressure field. This is done by mapping the element to a parent element and using the vertex pressures and element shape functions. Once a single pressure value is obtained for each pore, the hydraulic conductivity is computed using equation 44.

3.3.4.1 The problem with pressure

Of all the complications encountered during this work, one which required the most attention was reconciling the pressure fields in each model. Each pore in the network comprises a single pressure value. In contrast, the finite element model has numerous elements (and therefore vertices/pressure nodes) in each pore, so choosing a single value to represent the pressure field is challenging. This problem is not specific to FEM simulations – LBM and any DNS technique which discretizes the void space and solves for a continuous pressure field will face problems when trying to assign a single pressure value to be representative of the local pressure field.

The problem is compounded by the fact that neighboring pores can be merged together during the network generation process. Here, *merging* means that neighboring void regions would have had multiple unique voxel labels according to the locations of maximal inscribed spheres, but are all labeled with a single tag after being consolidated due to some user-specified merge criteria.
Commonly used merge criteria include overlapping of neighboring inscribed spheres and touching of neighboring inscribed spheres (more severe than overlapping). In either merging case, it is likely that some constriction likely exists within an identified pore. Therefore, the pressure gradient across the pore is likely to be non-negligible, and such that a single pressure will not be representative of all the voxels composing the pore (more so than in unmerged pores, where the pressure gradients across pore bodies are lower than across the surrounding throat constrictions).

Unfortunately, the problem cannot be solved by simply not merging at all. In a typical voxel image, there is a staircase effect which produces many burn maxima. Completely abandoning the pore merging step in the network generation process yields a large number of pores with unusual geometries and properties. Among these geometric problems, it is more likely that very small throat lengths exist, which lead to large numerical error when computing conductivities (pressure values can be nearly identical for both pores, causing unreasonably high values in $g_{ij}$ when flowrate is divided by pressure drop). Also, the throats between these pores are not the obvious interfaces at constrictions found in stereotypical pore throats. Rather, they are often jagged, pixilated surfaces which come into contact with several other throats (the reason this is a problem will become apparent later).

Adding to the problem of pore pressure determination is the fact that the pressure drop across each throat should correspond to the net flow across that throat face, but with each pore having multiple throats it is possible that a single pressure value could never satisfy all the pressure drop and flux relationships for a pore. A toy example to illustrate this problem is shown in Figure 3.16.

Although this is not an issue in the majority of pores, a more common scenario is that pressure interpolated at the location of the maximal inscribed sphere center is not a good value to represent the entire pore, but some other location would have been better. The first solution that likely comes to mind is handling these on a pore-by-pore basis: The hydraulic conductance is computed for all the throats within the meshed subdomain, and these problems present themselves as $g_{ij}<0$, which would mean fluid flows from low pressure to high. When a negative conductance is found, the pore pressures on each side of the throat are checked to determine which type of error is causing this: over-merged pore, or bad choice of interpolation location. If the latter, a new pressure is computed for that pore which will satisfy all the pore throats attached to it. This checking procedure involves (for each of the two pores) interpolating pressure at all the throat locations adjoining that pore to its neighbors. These provide upper and lower bounds on the pore pressure when combined with the sign of the corresponding throat flowrates. They also help to adjust the pore pressure to the proper value if necessary.
This leads to the question “What about when pore pressure values are bad, but don’t violate the pressure-flow inverse relationship? Couldn’t better conductivities be achieved by adjusting all the pore pressures?” However, it doesn’t make sense to adjust pressures in this manner. Once you start adjusting pressures to improve results you are no longer representing the physical flow field that was computed by the FEM simulation. Instead you are just imposing an arbitrary set of bounds that will guarantee pressure behaves a certain way. By “correcting” a pore’s pressure, we are using the information about the flow and pressure fields to compute $g_{ij}$ when $g_{ij}$ should be only a function of the pore geometry. This will lead to additional data leakage. Also, we do not want the neighboring throats to affect a throat’s $g_{ij}$. One might argue that the INP method discussed in 3.3.4.1.3 does allow neighboring throats to influence the pressure drop and therefore $g_{ij}$. In that case the pressures of the entire network are allowed to equilibrate within some constraints, as opposed to nudging a single pore’s pressure to achieve some ideal. In the context of striving to produce a PN which computes the same throat flowrates as the FEM model for a given set of boundary conditions, using INP is the most obvious way to achieve this.

Upon visual inspection of the velocity and pressure fields, it was realized that these pressure-flow disagreement problems tend to occur when the local net flow direction is nearly parallel to a throat’s face. Often, changing the overall flow direction (e.g. from $+z$ to $+x$) will alter the local flow fields enough to be more normal to the throat interface. Therefore, a methodology was adopted in which the models were run three consecutive times, one for each flow direction. Then, for each throat $ij$, the simulation with the largest net flow from $i$ to $j$ was used to compute $g_{ij}$ as long as it did not yield a negative value.

In step 9 of the workflow, the choice of pressure to represent the pore is somewhat arbitrary. While there is no guarantee that the value interpolated at the center of the inscribed sphere will be representative of the entire pore, it is arguably as good a choice as any for maintaining a consistent definition. Another more physics-based way to find a representative pressure for the pore is to use...
the pressure at the location of minimum pressure gradient within the pore. To determine this point, the 2nd-order accurate centered finite differences were used.

The idea of connecting the FEM and network models is centered on adjusting network $g_{ij}$ values in throats so they reflect the true flow resistance, making fewer broad assumptions about the uniformity of pore/throat geometries throughout the material. To get new $g_{ij}$ values for throats we use the finite element (or any streamline-scale) model to compute accurate velocity and pressure fields, integrate velocity over area for each throat’s flowrate, then back out $g_{ij}$ based on the pressure drop over the throat. However, the pressure drop across a throat requires a discretization of the FEM pressure field that is not obvious. In the network, each pore has a single value of pressure which dictates the pressure drops over all throats connected to it. In contrast, the FEM model provides a continuous pressure field throughout the entire volume of the pore. Because the pressure drop is essential to the $g_{ij}$ calculation, several different approaches were taken to determine a single FEM pore pressure for each that would be representative of the whole pore. Three of these approaches will be discussed in the following sections. They will be referred to as Pore Location (PLP), Minimum Gradient (MGP), and Iterative Network (INP) Pressures.

### 3.3.4.1.1 Pore Location Pressure (PLP)

First, the pore location from the network model was used. This location is defined as the center of the largest inscribed sphere. It is found by using Powell’s method to find a local maximum in the objective function: minimum distance to the nearest solid surface. This is done during the network generation process, before overlapping pores are merged. When two pores are merged due to having their inscribed spheres overlap, the location of the larger pore is used. To get the single value of pressure that will represent the pressure field with a pore using PLP, the FEM pressure solution is interpolated at the pore location using the linear shape function.

Though it is geometry-based, this method relies on a somewhat arbitrary choice of pore pressure, and it is possible that the problems mentioned above can occur. It does benefit from requiring no additional computation, and maintains a definition of the pore that is consistent with the network. Also, it makes sense intuitively that this would be an acceptable choice: the majority of the pressure drop between two pores takes place near the constriction of the throat, and the pore centers are typically far enough from the throat location to not be within the area of highest pressure drop. The center of the pore’s inscribed sphere is a place where the pressure gradient should be relatively small in magnitude. This idea leads us into the next method, MGP.

For simple cubic sphere packs, PLP should yield identical results to the other methods since the pores are symmetric and geometrically consistent, and network pore anomalies are not likely.
3.3.4.1.2 Minimum Gradient Pressure (MGP)

This approach is a physics-based way of choosing which point to interpolate the pressure field at. Using the 2nd order centered finite difference method, the minimum of the magnitude of the local pressure gradient is found by following the direction of the 2nd partial derivative.

\[
\frac{\partial P}{\partial x} = \frac{P_{NW}(x+1,y,z) - P_{NW}(x-1,y,z)}{2h}
\]

\[
\frac{\partial P}{\partial y} = \frac{P_{NW}(x,y+1,z) - P_{NW}(x,y-1,z)}{2h}
\]

\[
\frac{\partial P}{\partial z} = \frac{P_{NW}(x,y,z+1) - P_{NW}(x,y,z-1)}{2h}
\]

\[
|\nabla P| = \sqrt{\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2}}
\]

This is done to tie the physics of the two models together: the local minimum is a saddle point, where the flow field “splits”. Using MGP results in a slight improvement in throat pressure drops (and \(g_{ij}\)) compared to PLP in some cases. This gives some added confidence in the use of simpler PLP method. It suffers from many of the same drawbacks, such as requiring three simulations to test multiple flow directions. Also, there can be multiple local minima within the volume of a pore, especially when the pore merging criteria is more liberal (more merges). It is assumed that the minima nearest to the pore location is the best to use, though this wouldn’t always be the case. For this reason, this method does not guarantee agreement between the flow and pressure fields. In other words, it can still result in negative \(g_{ij}\) in some cases. Another drawback of this method is that because the pressure interpolation location is no longer tied to the PN pore locations, the PN throat length does not correspond to the distance between FEM pressure interpolations. This problem will lead to outliers when the distance between minima is significantly different than the distance between pore centers. Instinct suggests we could improve the model by simply adjusting the network throat length accordingly, but this would be another case of data leakage (where something is known in the training and test data which won’t be
known in the features of future predictions). We will maintain original throat lengths to avoid reporting unrealistically low prediction errors.

3.3.4.1.3 Iterative Network Pressure (INP)

To ensure a value of conductivity is backed out for each meshed throat, the INP method was developed. This approach does not use the FEM pressure field for interpolating pore pressures. Instead, the network pressures are used along with the FEM flowrates for computing $g_{ij}$ values in each iteration. The idea behind this approach is that in order to properly couple the two models, there needs to be more feedback beyond the application of network boundary conditions on the initial FEM mesh (which is arguably unnecessary for computing hydraulic conductances of throats, but is done for the purpose of producing physically realistic flow fields). In the PLP and MGP approaches, the coupling is sequential or hierarchical in nature (the FEM solution does not really need any information from the network to compute its solution, so it can be run a priori in a "parameter passing" type of workflow.) That is computationally advantageous, but suffers from a lack of congruence. On the other hand, one can minimize error between flowrates in each model by iteratively computing conductivities using network pore pressures and FEM flow rates. The following procedure is illustrated in Figure 3.17.

1. Once a mesh and PN have reconciled connectivities, the network solver computes pressures for each pore and flowrates for each throat.
2. The PN pressure solution is used to apply boundary conditions to the Neumann-labeled element faces of the mesh at the throat locations.
3. The FEM Stokes simulation is run, and the velocity field is integrated over throat faces to determine FEM throat flowrates.
4. The PN flowrates are compared to the FEM flowrates to determine if the desired RMSE tolerance has been met.
5. If the tolerance has not been met, the PN pressures and FEM flowrates are used to compute the $g_{ij}$ for the next iteration and steps 1-5 are repeated.
6. Once tolerance has been met, the converged hydraulic conductivities are output.
A benefit to this method is that no assumptions must be made about the pressure value which represents the continuous pressure field within each pore. Rather, this is automatically handled by the network. This is particularly useful in cases where interpolation at a single physical location cannot yield a representative pore pressure, such as when pores are over-merged, resulting in significant pressure gradients within the dual-pore’s volume.

The INP method can be expensive time-wise, depending on the number of iterations required to reach convergence, but the computational requirements are minimized by reusing the factorized sparse-stored stiffness matrix of the FEM model in each iteration, and only updating the RHS after each new network solution is obtained. For this reason, the memory requirement is approximately the same as the other methods. In one instance, using MGP required approximately 11 Gb of RAM and 20 minutes while the same hybrid model required 13 Gb and 80 minutes using INP with 10 iterations.

Because local mass balance is not explicitly enforced by the Galerkin FEM with Taylor Hood elements, convergence of the two models’ throat fluxes is not guaranteed, but has been observed in test cases (Figure 3.18).
I have considered there may be a possibility of data leakage in the calculation of the training examples when using the INP method of pressure drop calculations. When iterating back and forth between network and FEM, the starting point of each pore pressure does seem to have an impact on the final conductivity values which the algorithm converges to. Therefore, it matters which conductivity model is used for the initial network solution. For example: when using the Hagen-Poiseuille equation to compute $g_{ij}$ for each throat while obtaining the initial pressure distribution, features based on the Hagen-Poiseuille equation may have unrealistically high predictive power.

Imagine a scenario where there are four pores connected in series (A→B→C→D), where no additional throats are connected to the central pores (B and C). The INP algorithm will adjust the conductivities between all neighboring pores $i$ and $j$ to $g_{ij} = Q_{ij}^{\text{FEM}}/(P_{i}^{\text{NW}}-P_{j}^{\text{NW}})$ in each iteration. However, since the flows through the three throats ($Q_{AB}^{\text{FEM}}, Q_{BC}^{\text{FEM}}, Q_{CD}^{\text{FEM}}$) are the same, the conductivities ($g_{AB}, g_{BC}, g_{CD}$) will only be scaled, and the distribution will remain. In other words, if the initial conductivities were $[g_{AB}, g_{BC}, g_{CD}]=[10, 10, 10]$, network pressure drops were $[\Delta P_{AB}^{\text{NW}}, \Delta P_{BC}^{\text{NW}}, \Delta P_{CD}^{\text{NW}}]=[2, 4, 6]$, and FEM flowrates were $[Q_{AB}^{\text{FEM}}, Q_{BC}^{\text{FEM}}, Q_{CD}^{\text{FEM}}]=[4, 4, 4]$, then the new conductivities would be $[g_{AB}', g_{BC}', g_{CD}']= [2, 2, 2]$. Notice this does not give us any new information about each throat’s conductivity relative to each other. It could be that most of the pressure drop occurs in throat AB in the FEM solution, while BC and CD are highly conductive. Based on the final results, in some cases the conductivity method was only off by a scaling factor, when really that model had low predictive power on that section of the network.

This source of data leakage has been addressed by initializing the hydraulic conductivities with an additional FEM simulation, using one of the other pressure determination methods (MGP or PLP). This way the pressure field within the subdomain is not biased by...
any of conductivity approximation model. In the initial simulation, artificial boundary conditions are applied to the FEM mesh by assuming a linear spatial distribution of pressures.

There are two reasons for using the INP method: to attempt forcing the two models into agreement, and to compute hydraulic conductivities when the representative pore locations are unknown. The latter will become clearer in the next chapter, where particles trapped in throat constrictions can encompass the center of the maximum inscribed sphere (PN pore location) and create multiple pressure gradient minima within the pore.

Until a more robust meshing routine and mass conservative FEM formulation are implemented, this method will not outperform MGP or PLP for quality data generation when they are appropriate. Whereas those methods do their best with each throat and simply discard data which violate some quality criteria (pore mass balance error tolerance, minimum pressure drop, etc.), INP must use all data points to maintain consistency between models and noise removal must be performed on the results. In the current implementation, this yields larger, noisier training sets.

3.3.5 Prediction and Validation

The set of known hydraulic conductivities is then combined with the corresponding PN-derived features to form the full set learning data, which is exported to software packages for statistical analysis (Matlab, JMP). The features taken directly from the PN include:

- Throat Inscribed R
- Throat Surface Area
- Throat Cross-Sectional Area
- Throat Perimeter
- Throat Wetted Perimeter
- Throat Shape Factor
- Pore i Coordination Number
- Pore j Coordination Number
- Pore j Coordination Number
- Pore i Inscribed Radius
- Pore j Inscribed Radius
- Pore i Volume
- Pore j Volume
- Pore i Surface Area
- Pore j Surface Area

To maintain consistency and produce the same result for $g_i$ and throat $g_j$ (conductance of the same physical throat, identified two ways in the PN), the pore with the larger inscribed radius is given the "i" designation, while the smaller is labeled "j". Features computed based on domain knowledge include:

- Ratio of Throat Wetted Perimeter to Total Perimeter
- Inverse Throat Length (defined as pore-to-pore distance)
- Inverse of Throat Length (defined as pore-to-throat-to-pore distance)
- Ratio of inscribed throat radius to inscribed radius of larger pore
- Throat Equivalent Radius/Length
- Hagen Poiseuille relationship using inscribed radius
- Hagen Poiseuille relationship using effective radius
- Hagen Poiseuille relationship using equivalent radius
- Hyperbolic venturi approximation using effective radius
- Aperture approximation
- Creeping flow solution through sinusoidal tube using inscribed radius
- Creeping flow solution through sinusoidal tube using effective radius
- Perturbation solution
- Mapping onto triangular tube
- Mapping onto rectangular tube
- Mapping onto elliptical tube
- Shape-factor based mapping to triangular cross-section
Exploratory data analysis is then done using JMP to identify outliers which should be examined, and which data points constitute noise rather than an underlying relationship. A machine learning model can then be trained on the data and used to make hydraulic conductivity predictions for the entire PN. The features which were used for training are extracted from every throat in the PN. For linear regression models, prediction is simply a matter of multiplying the throat features by the learned coefficients. Any transformation made to the predictors while training must be made in the same way to these features before this calculation. This means that normalized features must be scaled using the variance, mean and standard deviation of the training set. The array of predicted $g_{ij}$ values for each throat is sent into the single-phase flow PN model, and the prediction expression is saved as a function which can be called for future predictions.

3.4 Results and Discussion

3.4.1 Mesh Dependency Analysis

Prior to multiscale modeling, it must be verified that the mesh is sufficiently dense so that the sub-pore-scale results can be trusted. In order to keep the computational costs as low as possible while still maintaining accurate local flowrate calculations, the sensitivity to flowrate on mesh resolution was analyzed for a subdomain of each material of interest (Figure 3.19, Figure 3.20). Tests were also done on a computer-generated random sphere pack (Figure 3.19), pushing the resolution finer and finer until the simulation required more than the 96 Gb RAM available on the “bigmem” nodes of Philip HPC queue at LSU. The results were used to decide on the best mesh resolution and subdomain size for efficiently generating sub-pore-scale data. The number of elements is stated to provide a relatable idea of the amount of discretization, but that number’s relation to computational requirements is not the same for all materials. The memory is more directly related to the number of equations being solved for. For a given resolution high porosity materials have a larger ratio of equations to elements compared to low porosity materials because they have a smaller ratio of no-slip Dirichlet nodes to elements. Similarly, a high-resolution mesh of a small subdomain will require more memory than a low-resolution mesh of a large subdomain if they both have the same number of elements. This demonstrates the need for refinement near surfaces, where better conformance to the solid surface can be achieved without incurring prohibitive computational demands.

Previously, sensitivity of permeability or total flowrate through the full system was used to determine whether the results were mesh dependent. In the context of this work, however, the local flowrates are of primary concern and must be taken into consideration for this analysis. It is necessary to perform this mesh sensitivity analysis on at least every new combination of material type and image resolution, if not a portion of every image to be analyzed.
Figure 3.19 Left: Sensitivity of local flowrate computation on mesh resolution for Random Sphere Pack subdomain consisting of 59 meshed throats. RMSE and MAE are defined in terms of the throat flowrates ($Q_{ij}$) of the finest/densest mesh, which is why they are exactly 0 in the last data point. Right: RAM requirement for increasingly refined meshes. Mesh refinement tuning parameters are provided in Table C.2. Note that this memory-element relationship is specific to the sample. Memory depends on the number of equations to be solved for, not the number of elements.
Figure 3.20. Mesh sensitivity results for subdomains of CT images. Data are shown for the mean throat flowrate magnitude and mean absolute error, MAE, with meshes of increasing resolution. MAE is defined in terms of the $Q_{ij}$ of the finest/densest mesh, which is why they are always 0 in the last data point.

3.4.2 Predicting Hydraulic Conductivity

Learning algorithms perform best when large amounts of data are provided. A single meshed subdomain typically does not contain enough throats to properly train a model. Therefore, for each CT image, the following procedure was followed to generate training data. The full-domain image was used to extract a pore network model and the corresponding labeled poremap used for meshing. Then, using the information provided by the mesh sensitivity study, subdomain ROI size was chosen. The number of FEM subdomain meshes generated for the image was based on how many non-overlapping ROIs could fit in the domain. In an automated procedure, these subdomains were extracted, meshed, labeled, and assigned boundary conditions based on the PN single-phase flow solution. This was done using the LSU HPC Philip queue, so several the nodes could be used (and multiple simulations could be run)
simultaneously. Table 3.1 shows the sizes of the PN and FEM models and simulation time and memory requirements. The number of data points obtained for machine learning are also shown, listed as total number of usable meshed throats. This is the number of throats which provided a trustworthy flowrate and pressure drop, not the total number of meshed throats. Throats connected to the subdomain boundaries were not used since only one of the two pores would be meshed. Throats with extremely low pressure drop were excluded since these are typically dead-end or isolated porosity. Others were excluded due to poor mass balance. The total number of elements and total number of throats tell how many elements were used per throat on average: 6097, 4638, 2045, and 3287 elements per throat for random pack, glass beads, sandstone, and carbonate, respectively.

It should also be noted that the times shown in the table are for the completion of the coupled multiscale model, which actually consists of three sets of simulations, where the PN boundary conditions are applied on the outside of the full domain for pressure drop in x, y, and z. This is done to decrease the chances of having unreliable local throat pressure drops, as discussed in 3.3.4.1. The memory and time in the table are for the MGP method of pressure interpolation.

Table 3.1. Pore network and FEM subdomain sizes for each porous medium

<table>
<thead>
<tr>
<th></th>
<th>Random Sphere Pack</th>
<th>Glass Beads</th>
<th>Bentheimer Sandstone</th>
<th>Estaillades Carbonate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size (voxels)</td>
<td>500^3</td>
<td>1200^3</td>
<td>1000^3</td>
<td>650^3</td>
</tr>
<tr>
<td>Pores in PN</td>
<td>1,306</td>
<td>18,400</td>
<td>68,746</td>
<td>11,707</td>
</tr>
<tr>
<td>Throats in PN</td>
<td>10,196</td>
<td>45,903</td>
<td>94,974</td>
<td>21,276</td>
</tr>
<tr>
<td>Number of subdomains</td>
<td>8</td>
<td>125</td>
<td>125</td>
<td>27</td>
</tr>
<tr>
<td>Total Usable Meshed Throats</td>
<td>814</td>
<td>5,029</td>
<td>42,413</td>
<td>9,040</td>
</tr>
<tr>
<td>Avg num elements/subdomain</td>
<td>620,372</td>
<td>186,612</td>
<td>699,307</td>
<td>612,282</td>
</tr>
<tr>
<td>Avg memory required (Gb)</td>
<td>30.4</td>
<td>6.9</td>
<td>23.9</td>
<td>17.3</td>
</tr>
<tr>
<td>Avg time required (h:m:s)</td>
<td>0:29:00</td>
<td>0:07:14</td>
<td>0:45:41</td>
<td>0:34:08</td>
</tr>
</tbody>
</table>

Total run times were longer for the INP method, though memory was similar. INP times depend on number of iterations; one glass bead coupled subdomain model using INP for 10 iterations (totaling 30 coupled simulations) required 6.2 Gb and nearly 30 minutes to reach convergence. Currently, the bottleneck in the total workflow is in the meshing, which took over one hour for the refined meshes of sandstone and carbonate. Splitting into more smaller subdomains would lessen the costs of the sandstone and carbonate if more computational nodes were available for use simultaneously.

Ordinary least squares (OLS) regression was used for quick, easily-automated, and interpretable models for gij which can offer moderate improvement over the models from literature which were most appropriate to the material in terms of prediction accuracy. Using OLS is good for getting a feel for the data through exploratory statistical analysis, and framing the expectations of more advanced regression models. The results of OLS are shown in the following figures and comparisons are made between these OLS predictions and conductance models from literature before more advanced machine learning algorithms were implemented.
The data obtained for each material was split into training and test sets. The learning curves (Figure 3.21, Figure 3.22, Figure 3.24, and Figure 3.26) for each material illustrate how each type of error changes as more observations are added to the training set, with the test set size held constant. The percentage of data held out in the test set for each was 25%, 30%, 50%, and 40% for the random pack, glass beads, sandstone, and carbonate, respectively. This was varied because the amount of total data available was varied.

Here, the data was split randomly. As mentioned previously, it is recommended to stratify the data to avoid imbalancing the sets (or better, to use cross-validation techniques). This was done in later models, but these were run on random samples of raw data. The curves imply that the amount of training data was sufficient for all cases and additional data will not benefit the OLS model. In the sandstone, it seems we generated far more data than was needed, suggesting fewer (and/or smaller) subdomains could have been used. This is helpful as it would lessen computational costs. Inspection of the parity plots combined with the closeness of the converged test and training errors suggests that overfitting has not occurred. The magnitude of the converged test error is not quite the true error of the OLS model since there is still noise present, but it represents the variance in the data which cannot be explained by the model being implemented. Had the training error been drastically lower than test error, it would point to a case of overfitting to the training data and thus not generalizing well to new observations. Overfitting did not seem to be a problem in the three OLS models. The parity plots (Figure 3.23, Figure 3.25, and Figure 3.27) show the predictions of test set and training set observations in relation to the FEM $g_{ij}$. The learned expression from OLS for each material can be obtained from Table 3.2, which contains the coefficients for each feature used.

In the results for below, plots labeled “MGP$_{340}$ Predictions” signify that the minimum gradient pressure was used for interpolating the FEM pressure when computing true conductances and method 340 was used in the full domain PN to determine boundary conditions applied to the subdomains (Method 340 is code’s internal label for the formula for creeping flow through a sinusoidal throat (Sisavath, Jing et al. 2001), using inscribed radii. It was chosen arbitrarily). It was observed that the method used for external PN throat conductances had little impact on the data quality or values, though full sensitivity to this has not been studied. MGN was used in the real materials because it resulted in slightly cleaner data than PLP. In the Random Pack, PLP was used.
Figure 3.21. Random Pack Least Squares Regression Results for PLP data. Note: Units are cm$^3$, but the image is artificial, so magnitude of values is arbitrary.
Table 3.2. Multivariate Least Squares Regression Coefficients.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random Pack</td>
</tr>
<tr>
<td>Intercept</td>
<td>-323.660</td>
</tr>
<tr>
<td>i j Inscribed R</td>
<td>112.553</td>
</tr>
<tr>
<td>i j Surface Area</td>
<td>-0.191</td>
</tr>
<tr>
<td>i j Cross-Sectional Area</td>
<td>-1.139</td>
</tr>
<tr>
<td>i j Inverse Throat Length</td>
<td>-36.963</td>
</tr>
<tr>
<td>i j Perimeter</td>
<td>10.831</td>
</tr>
<tr>
<td>i j Wetted Perimeter</td>
<td>-9.999</td>
</tr>
<tr>
<td>i j Shape Factor</td>
<td>60.097</td>
</tr>
<tr>
<td>Pore i CN</td>
<td>-5.250</td>
</tr>
<tr>
<td>Pore j CN</td>
<td>-2.031</td>
</tr>
<tr>
<td>i Inscribed Radius</td>
<td>-14.981</td>
</tr>
<tr>
<td>j Inscribed Radius</td>
<td>10.331</td>
</tr>
<tr>
<td>i Volume</td>
<td>0.001</td>
</tr>
<tr>
<td>j Volume</td>
<td>0.000</td>
</tr>
<tr>
<td>i Surface Area</td>
<td>-0.017</td>
</tr>
<tr>
<td>j Surface Area</td>
<td>-0.008</td>
</tr>
<tr>
<td>i j Wetted/Total Perimeter</td>
<td>582.366</td>
</tr>
<tr>
<td>i j Inverse Length (i to ij to j)</td>
<td>-26.317</td>
</tr>
<tr>
<td>Rij/Ri</td>
<td>-1190.594</td>
</tr>
<tr>
<td>i j Equivalent Radius/Length</td>
<td>-465.845</td>
</tr>
<tr>
<td>EQ1. Hyperbolic Venturi, Reff</td>
<td>-1.803</td>
</tr>
<tr>
<td>EQ2. Hagen Poiseuille with Req</td>
<td>-10.315</td>
</tr>
<tr>
<td>EQ3. Hagen Poiseuille with Rinsc</td>
<td>-10.393</td>
</tr>
<tr>
<td>EQ4. Hagen Poiseuille with Reff</td>
<td>1.178</td>
</tr>
<tr>
<td>EQ5. Ball and stick model</td>
<td>-75.697</td>
</tr>
<tr>
<td>EQ6. Map to elliptical cross section</td>
<td>44.964</td>
</tr>
<tr>
<td>EQ7. Aperture approximation</td>
<td>-1.339</td>
</tr>
<tr>
<td>EQ8. Sinusoidal tube with Rinsc</td>
<td>0.594</td>
</tr>
<tr>
<td>EQ9. Sinusoidal tube with Req</td>
<td>3.675</td>
</tr>
<tr>
<td>EQ10. Perturbation solution</td>
<td>-0.113</td>
</tr>
<tr>
<td>EQ11. Map to arbitrary triangular tube</td>
<td>42.465</td>
</tr>
<tr>
<td>EQ12. Map to rectangle</td>
<td>-6.021</td>
</tr>
<tr>
<td>EQ13. Shape Factor map to triangle</td>
<td>1.114</td>
</tr>
</tbody>
</table>
Figure 3.22. Glass Beads learning curve. Convergence suggests acquiring additional data would not be helpful or necessary. Nearly identical test set and training set errors give confidence that overfitting has not occurred.

Figure 3.23. Glass Bead Parity Plot: OLS performed on training data (blue). Test set used for error reporting and validation.
Figure 3.24. Benheimer Sandstone learning curve. Convergence suggests acquiring additional data would not be helpful or necessary. Similar test set and training set errors give confidence that overfitting has not occurred.

Figure 3.25. Bentheimer Sandstone Parity Plot: OLS performed on training data (blue). Test set used for error reporting and validation. Units are cm$^3$. 
Convergence suggests acquiring additional data would not be helpful or necessary. Similar test set and training set errors give confidence that overfitting has not occurred.

The noise present in the data is due to a combination of factors discussed in 3.2.6.5.2, some of which are drawbacks of the simplification of network modeling, while others could be lessened by improvements to the PN extraction and FEM meshing.
algorithms. Looking at the worst outlier in the Bentheimer Sandstone predictions (Figure 3.25, bottom right red data point: FEM $g_{ij} = 2789$; OLS $g_{ij} = 179.5$), a couple of obvious things are causing large prediction error. The ratio of wetted perimeter to total perimeter is very low ($\frac{P_{wet}}{P_{total}} = 0.078$), meaning this throat is surrounded by other pores. So, flow through this throat will have very little resistance (but there could be a significant $\Delta P$ due to the surrounding pores). This is why the models will consistently underestimate conductances of throats exhibiting low wetted perimeter ratios (the highest value of all the approximation formulas tested was 67.3). Looking at the poremap, there was also a spoke of another pore cutting through the center of the throat, which could be causing it to have strange properties. This is probably caused by the merging of the neighboring pores in vox2net, but the max number of merges was reached before one of these could be merged into the spoked pore. A 2D cross-section of the poremap cutting through the throat and both pores are shown in Figure 3.28.

Figure 3.28. 2D cross-section through poremap near throat ij which caused severe outlier in prediction data (pore i is pink, pore j is purple); the spoke is the yellow voxels in the middle of the throat (spoke is perpendicular to the image)

The sensitivity of the data quality to FEM pressure interpolation method (INP, MGP, PLP) was tested and revealed similarities between PLP and MGP, which gives confidence in using interpolated FEM pressures if a consistent definition of pore location is maintained. The data from INP stood out as being noisier (Figure 3.29). Further investigation showed that convergence of INP varied greatly amongst subdomains, causing the merged datasets to exhibit a lot of noise relative to the other methods (Figure 3.30).

Convergence was defined in terms of MSE between FEM and PN throat flowrates. While initial tests of the INP algorithm showed promise for convergence between the two models, there are still issues in the way of guaranteed convergence between them. These problems can be solved on a case by case basis, but systematically it is still a challenge that requires improvements to the meshing and PN extraction methods mentioned previously and discussed in 5.1. One of these was identified through visualization as disconnected portions of meshed pores which do not show up in connectivity checks because the meshed pore is split into multiple
clusters of elements. This is illustrated in Figure 3.31, where a flat, relatively long throat was not resolved by increasing refinement in throat. A larger extent of refinement would be required to ensure the mesh is dense enough throughout the throat. Default extent is the inscribed radius of the throat.

![Graphs showing comparison between Glass Beads - INP Predictions, Glass Beads - MGP Predictions, and GB PLP Predictions.](image)

Figure 3.29. PLP, MGP, and INP Comparison: OLS predictions on test and training data reveal significantly less noise in PLP (bottom) and MGP (middle) data. OLS performs poorly on the more noisy data from INP (top). Units are cm³.
Figure 3.30. Histogram of INP MSE at end of 10 iterations. MSE defined in terms of throat flowrates in FEM and PN models. 125 subdomains using INP method of pressure determination for conductance calculation.

Figure 3.31. Top: example 2D illustration of closed mesh which does not violate connectivity agreement between models (yellow pore and green pore are connected in both mesh and PN), but will make resolving flowrates impossible. Blue arrow points to an element associated with the yellow pore which causes the two pores to technically be connected in the mesh, but the flow across the interface can only be due to recirculation in the dead-end pore space. Bottom: identified region where this is taking place.

A somewhat unexpected result was the similarity amongst the three materials with respect to traditional conductance calculations. One might suppose these drastically different materials would each be best-suited to different $g_{ij}$ estimation formulas. Each method performed relatively similarly. This suggests that the methods which consistently perform best are best-suited for the network extraction technique and associated seeding and merging options, rather than these materials specifically. It is likely that the trends would look quite different for a pore body-channel network. Figure 3.32 shows the test set prediction performance (RMSE) for the
OLS prediction compared to several conductance models from literature. The error is with respect to the FEM-computed conductances using MGP.

Figure 3.32. Histogram Comparison of OLS learned models (first column of each) against conductance formulas found in literature show OLS consistently best, if only slightly. Left: Glass Beads. Center: Bentheimer Sandstone. Right: Estaillades Carbonate. Bin labels are associated with the in-house numbering system used in the source code.

Figure 3.33. Histogram for error comparison of various models on Random Pack data. OLS significantly outperformed traditional methods. Relative performance of other models also differed from the three real materials.

Figure 3.34 compares the predictions of the OLS model to the hyperbolic venturi (HV) approximation method, which consistently performed best amongst the traditional methods. There are a few negative predictions made by the OLS model which can be easily identified and adjusted. The adjustment is necessary because negative conductance is nonphysical. When negative conductances are encountered, we (arbitrarily) set them equal to 10% of the minimum nonnegative predicted value. The learned model outperforms the traditional one in terms of RMSE, MAE, and $R^2$, even before those adjustments. Clearly, the hyperbolic venturi consistently underestimates the conductivity.
While this looks like a very successful result, we are interested in outperforming traditional models both in terms of total error and distribution, and a large amount of the HV error seems to be due to the model being off by a scaling factor. So, it is necessary to scale the HV approximation conductivities to match the FEM permeability. Doing so cut the prediction error in half and provided a clear picture of how the distributions vary between the models (Figure 3.35). The tighter distribution of the OLS values (quantified by the nearly 50% lower RMSE) around the target line compared to the HV gives us confidence in comparing the relative errors.
Figure 3.35. Comparison of conductivity distributions. Same data as Figure 3.34, but only showing held-out test data, and the Hyperbolic Venturi approximation has been scaled to match the permeability of the sample. The spread of the scaled values (green) is wider than that of the OLS (blue). Units are cm$^3$.

Forward selection and lasso were also used to obtain sparse expressions for random pack throat conductivities. Both methods performed very similarly on this data. Figure 3.36 shows the solution path and Figure 3.37 shows the predictions using the sparse model from forward selection, which included 14 features.

Figure 3.36. Forward selection solution path. Parameter estimates (left) show the magnitude of coefficients, which tend to be very high in overfit models with collinear features. Right: -Log Likelihood describes the plausibility of the parameter values, which should be low. The solid red vertical line is step used in Figure 3.37 and equation 50.
We note that the issues associated with the ratio of wetted to total perimeter exhibited by the Bentheimer outliers (Figure 3.28) do not affect the random pack data. Figure 3.38 shows the distribution of this feature for the random pack, and very few throats have ratios below 0.6, hence the very clean data.

\[ g_{ij}^{FS\text{ (Random Pack)}} = -990.2 - 0.2352(\text{Surface Area}) + 15.04(\text{Perimeter}) - 14.64(\text{Wetted Perimeter}) + 7518.2(\text{Shape Factor}) - 7.325(i\text{ Coordination Number}) + 9.942 \times 10^{-4}(i\text{ Volume}) - 0.01997(i\text{ Surface Area}) + 0.01375(j\text{ Surface Area}) + 1293.1 \left( \frac{\text{Perimeter}_{\text{Wetted}}}{\text{Perimeter}_{\text{Total}}} \right) - 10263.2\left( \frac{\text{Length}_{i\text{ to } j\text{ to } f}}{R_{ii}} \right)^{-1} + 87.69 \left( \frac{R_{ii}}{R_{ij}} \right) - 1.181(EQ3) + 1.636(EQ9) - 0.4759(EQ13) \] (50)

Figure 3.37. Random Pack Training and Validation set performance using forward selection expression (50). Units are cm³.

Figure 3.38. Random Pack Property Distribution: Ratio of wetted to total perimeter
Table 3.3. Coefficients for linear regression algorithms with feature selection for sparse models on the Bentheimer MGP data set. RMSE (on test set, with respect to FEM $g_{ij}$) and $R^2$ reported for comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>Bentheimer Sandstone Coefficients</th>
<th>Elastic Net $\alpha = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Forward Selection</td>
<td>Lasso</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.883</td>
<td>0.882</td>
</tr>
<tr>
<td>Test RMSE</td>
<td>88.0</td>
<td>85.78</td>
</tr>
<tr>
<td>Number of Parameters</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>Intercept</td>
<td>14.51597</td>
<td>0.4506533</td>
</tr>
<tr>
<td>$ij$ Inscribed R</td>
<td>-1.55571</td>
<td>0</td>
</tr>
<tr>
<td>$ij$ Surface Area</td>
<td>-0.01047</td>
<td>0</td>
</tr>
<tr>
<td>$ij$ Cross-Sectional Area</td>
<td>0.581793</td>
<td>0</td>
</tr>
<tr>
<td>$ij$ Inverse Throat Length</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$ij$ Perimeter</td>
<td>0</td>
<td>0.3045913</td>
</tr>
<tr>
<td>$ij$ Wetted Perimeter</td>
<td>-0.643695</td>
<td>-0.2803729</td>
</tr>
<tr>
<td>$ij$ Shape Factor</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pore $i$ CN</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pore $j$ CN</td>
<td>0.767367</td>
<td>0</td>
</tr>
<tr>
<td>$i$ Inscribed Radius</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$j$ Inscribed Radius</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$i$ Volume</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$j$ Volume</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$i$ Surface Area</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$j$ Surface Area</td>
<td>-0.00293</td>
<td>0</td>
</tr>
<tr>
<td>$ij$ Wetted/Total Perimeter</td>
<td>-5.5292</td>
<td>0</td>
</tr>
<tr>
<td>$ij$ Inverse Length ($i$ to $ij$ to $j$)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$R_{ij}/R_i$</td>
<td>-16.4764</td>
<td>0</td>
</tr>
<tr>
<td>$ij$ Equivalent Radius/Length</td>
<td>7.713934</td>
<td>0</td>
</tr>
<tr>
<td>EQ1. Hyperbolic Venturi, Reff</td>
<td>0.597335</td>
<td>0.1248974</td>
</tr>
<tr>
<td>EQ2. Hagen Poiseuille with Req</td>
<td>0</td>
<td>-0.149385</td>
</tr>
<tr>
<td>EQ3. Hagen Poiseuille with Rinsc</td>
<td>0</td>
<td>0.1756082</td>
</tr>
<tr>
<td>EQ4. Hagen Poiseuille with Reff</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EQ5. Ball and stick model</td>
<td>0</td>
<td>-0.352397</td>
</tr>
<tr>
<td>EQ6. Map to elliptical cross section</td>
<td>0</td>
<td>-0.28618</td>
</tr>
<tr>
<td>EQ7. Aperture approximation</td>
<td>-0.60999</td>
<td>0</td>
</tr>
<tr>
<td>EQ8. Sinusoidal tube with Rinsc</td>
<td>1.214508</td>
<td>0</td>
</tr>
<tr>
<td>EQ9. Sinusoidal tube with Req</td>
<td>0</td>
<td>1.6413439</td>
</tr>
<tr>
<td>EQ10. Perturbation solution</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EQ11. Map to arbitrary triangular tube</td>
<td>0</td>
<td>0.3903823</td>
</tr>
<tr>
<td>EQ12. Map to rectangle</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EQ13. Shape Factor map to triangle</td>
<td>-0.09978</td>
<td>0</td>
</tr>
</tbody>
</table>

Sparse models were created using several of the techniques discussed in 3.2.6.1. Tuning parameters for each algorithm provide the capability to generate prediction expressions of varying complexity. Sparser models can have high bias but are more robust to noisy data, so this trade-off must be considered when selecting a model. Table 3.3 contains coefficients for sparse prediction expressions generated by forward selection, lasso, and elastic net. Comparing test set performance of sparse models against the full least
squares model showed that the following expression, learned using the lasso shrinkage method, exhibited a good bias-variance trade-off for the Bentheimer sandstone data.

\[
\begin{align*}
g_{ij}^{\text{Lasso}} &= 0.450653 + 0.30459 \text{(Perimeter)} - 0.28037 \text{(Wetted Perimeter)} + 0.1248974 \text{(EQ1)} \\
&\quad - 0.149385 \text{(EQ2)} + 0.1756082 \text{(EQ3)} - 0.352397 \text{(EQ5)} - 0.28618 \text{(EQ6)} \tag{51} \\
&\quad + 1.6413439 \text{(EQ9)} + 0.3903823 \text{(EQ11)}
\end{align*}
\]

The conductance approximations for the \textit{EQ} terms are computed using the formulas in APPENDIX D. This expression had a slightly lower test set RMSE than the OLS expression for the hyperbolic venturi approximation in terms of conductance and throat flowrates, but the small improvements had a compounded positive impact on total flowrate estimation. In order to make this comparison it was necessary to use a new domain where full DNS could take place. A \(250^3\) voxel cutout of the original image was discretized into a full PN model (866 pores, 1304 throats) and full FEM mesh (3.2 million elements). The FEM simulation was conducted first to determine the proper pressure boundary conditions on the outside of the PN and make the comparison valid. This was the maximum domain size which could be simulated on the 96 Gb node. Meshing required 3Gb RAM and over 4 hours. A single Stokes simulation required over 96 Gb and 2 hours (simulation time would have been less on a node with more RAM so no swapping of virtual memory occurred – reading from disk is much slower than reading from memory). The domain is near the characteristic length scale for porosity, but is too small for permeability (see Figure 3.39). Therefore, the full DNS approach is not valid for this image of this material, showcasing the need for the multiscale technique.

![Figure 3.39](image)

Figure 3.39. Characteristic Lengths for Bentheimer Sandstone in terms of porosity (left) and permeability (right). The domain size used for testing was \(250\) voxels (\(0.075\) cm). Converged values agree with published results (Muljadi, Blunt et al. 2016).

While we were hopeful that predictive expressions could be derived for each material, it was not anticipated that a single expression would have high predictive performance on other materials exhibiting considerably different properties. The success of the
implementing the lasso expression (trained on sandstone data) in the sandstone test domain, combined with the results in Figure 3.32 (where conductance models perform similarly for all three materials), seem to suggest that the lasso expression could be applied to the other materials. Using the entire carbonate data set as a test, the lasso expression was applied to the feature set and compared to the FEM-computed conductances resulting in a RMSE of 63.98, on par with the test set error of the OLS model trained on carbonate data (RMSE=64.40, Figure 3.27). See Figure 3.40.

![Estallades Carbonate Hydraulic Conductivity Prediction](image)

**Figure 3.40.** Prediction of carbonate throat conductivity using lasso expression trained on sandstone data.

### 3.5 Conclusions

A multiscale modeling framework was developed which couples finite element and pore network models at the pore-level using a heterogeneous multiscale method. Both models benefit from the coupling: Pore-scale boundary conditions applied to the FEM model circumvent the artificial confinement of fluid typically imposed and allow smaller subdomain meshes to provide useful information, while the PN assumption that all throats are well-represented by a single simplistic geometry is not needed. Hydraulic conductivity, the essential parameter in PN modeling which typically incorporates many oversimplifying assumptions, is learned from the FEM Stokes solution using modern regression techniques. Regression models trained on observations of FEM-computed conductances and PN-derived features were used to develop prediction expressions for porous materials.
For three real CT imaged porous materials, these techniques were successful in modeling pore-scale flow more accurately and with fewer broad assumptions than traditional methods. Computational requirements of the hybrid model are drastically lower than full DNS simulation of the full imaged domain, which would require more than the currently available resources at the required level of discretization. Still, challenges with meshing and network extraction prevented us from achieving the level of pore-level accuracy we hoped for and believe this technique is eventually capable of.

Comparisons were made between several methods from literature for computing hydraulic conductivity. The hyperbolic venturi approximation using the effective throat radii (Thompson and Fogler 1997) had the highest prediction performance compared to other traditional models. This was the case for three unique porous materials, suggesting the method is well-suited for the PN extraction technique employed throughout this study. This was not known prior to this study. Previously, methods were compared on the basis of total flowrate or permeability, and the hyperbolic venturi did not consistently outperform other methods. This shows the utility of the multiscale model in gaining insights into specific porous materials, and the pore-scale modeling techniques being implemented.

The fact that the OLS predictions did not always drastically outperform one of the traditional PN models in some cases does not indicate a problem with the formulation or implementation of the hybrid model, but a side effect of the noise in the data. Errors are reported on a held-out test subset of the data for honest comparison. The OLS model was run prior to extensive data cleaning and feature selection. For noisy data modeled using multiple correlated features (certainly the case here), it is common to find relationships in the noise. Some sources of noise have been identified during this study, and the most blatant outliers tended to exemplify multiple sources. There are small errors in FEM-computed flowrates associated with the Taylor-Hood elements which leads to error in pore-wise mass balances, especially for coarsely-meshed pores.

There is room for improvement in several areas: Network extraction can be improved by considering multiple throats between two pores, and developing new criteria for pore merging which will prevent anomalous geometries. Closer examination of outliers revealed difficulty in predicting conductances of throats with a low ratio of wetted to total perimeter. These pores are surrounded by other pores instead of solid phase, so the no-slip assumption (which is built into all the traditional Poiseuille-based models) is not upheld. The FEM flowrate through these throats are much higher than what is predicted by any of the traditional models or learned models developed here. The lasso expression in equation 51 shows that the shrinkage method attempted to recover some of this information, keeping the perimeter and wetted perimeter as important features.
Different methods were tested for computing the “true” conductivities of throats based on different ways of determining the pressure drop across a throat (PLP, MGP, INP). In the end, the original method of interpolating at pore centers tended to provide the cleanest data, especially in the case of the sphere packs. MGP produced very similar data to PLP and would be improved by updating the pore and throat locations used as features in ML, but we won’t have this information available to use in future predictions, so it would be a case of data leakage (falsey reported accuracy). INP requires better agreement between mesh and network structures before it can generate clean data sets. It has the advantage over the other methods of comparing the throat flowrates of the PN and FEM models, making it a more strongly coupled approach. It will be especially useful in cases where interpolation of the FEM pressure field is not possible, such as when a particle becomes trapped and covers the pore center.

It was determined that the best traditional conductivities models actually predict conductances reasonably well in most cases, so there is less room for improvement than initially expected. It seems the low-hanging fruit has already been taken: e.g., in the case of real geologic materials, the true error of the hyperbolic venturi approximation model is nonlinear with respect to the features we are using. I.E. including additional predictors in least squares multivariate regression yielded only small improvements over the original model. Using a more complex machine learning algorithm for prediction provided more improvement, making this approach worthwhile in cases where accuracy of pore-level predictions is vital. It is also recommended to use these techniques for validation of the appropriateness of the conductance model being used for a given network extraction technique. The results are perhaps not compelling enough to warrant the use of the multiscale methodology by those whose primary concern is macroscopic parameter upscaling and when individual models suffer from the same problems encountered in this work.

During the machine learning process, error is defined in terms of the hydraulic conductivities of each throat, which are geometrically-defined parameters and should be independent of the flow field. While throat flowrates are the essential flow results that we hope to accurately model, it is better to compare conductivities directly. Local pressures and velocities/flowrates are dependent on several non-local factors in both types of models (boundary conditions, non-local heterogeneities and structures, etc.) whereas conductivity is geometric, and independent of external influences beyond the local neighborhood of pores. Therefore, when analyzing throat-wise prediction errors, it makes sense to use values which are independent of one another to eliminate the compounding effect of a single error. In a tight porous media, with several distinct flow paths, a single flowrate anomaly in one flow path would impact the flowrates of all the other throats in that path. For example: imagine two disconnected flow paths through a small porous medium with five throats each. One of the flow paths has perfectly modeled throats, while the other has four perfect, one in error causing 50% overestimation of that paths flowrate based on the applied BCs. The error of one impacted the flowrates of the other four throats. It was thought that a proper comparison could be made by simply scaling all the throat flowrates by the total
flow through the system, but doing so will also scale the good predictions, making them worse. Therefore, throughout this study an emphasis was put on determining trustworthy “true” throat conductivities. While the improvement in prediction accuracy of the multiscale model over traditional models was lower than anticipated, the small improvements at the pore level can be significant in light of the compounding impact of errors on the fully interconnected network.

A key finding in the study was the apparent independence of conductivity prediction formula accuracy to the material being studied. Prediction expressions using data generated entirely from sandstone information was successfully used to compute conductivities in carbonate with the same level of accuracy as an expression learned from carbonate-derived data. This finding leads to another hypothesis, which has not yet been tested: Using this multiscale data generation process across many images of many types of porous materials and combining the data into a single set of training data could provide enough data for big data-level algorithms to learn deeper structures in the data.

Tree-based methods will be especially useful in future applications of the multiscale model when a good approximation of the underlying data isn’t already available. In the case of conductance prediction, several relationships have been proposed for relating flow to pressure and these can be used as terms in the regression model (the EQ terms in Table 3.3). Figure 3.41 shows results of a boosted tree using only the geometric features (no EQ terms) to predict conductivity of throats in glass bead packs. This proof-of-concept shows that the machine learning approach is not fully reliant upon powerful physical models of the underlying phenomena, although they are certainly helpful.

Figure 3.41. Boosted Regression Tree. Left and Center: First two layers of a 50-layer boosted regression tree used for predicting throat conductance from PN geometric parameters of glass bead pack. Right: Sample conductance prediction performance on validation data. R²=0.867, RMSE=219.04.
It should be emphasized that the prediction expressions provided here are non-unique. Tests have shown that adjusting the machine learning hyperparameters can result in drastically different models (with completely different non-zero predictor terms in some cases), but very similar prediction performance (RMSE and R²). This is due to several factors, including the correlation of several of the predictors used here.
4 MULTISCALE MODELING OF PARTICLE FILTRATION

4.1 Introduction

Particulate flow and filtration is an important area of study for engineers in multiple disciplines (water purification, contaminant remediation, separations, and reservoir formation damage). Keeping in mind the goal of a model capable of predicting how retention of particles impacts continuum-scale parameters, physically-representative pore network (PN) and finite element method (FEM) models are used together for simulating straining phenomena during particulate flow in saturated porous media. A framework was developed for generating data which quantifies the impact of intercepted particles of varying size on the flow resistance of throats with varying geometry. Using supervised machine learning with FEM-computed hydraulic conductivities (before and after straining) and network properties serving as the training set, models are trained for predicting hydraulic conductivity reduction. These trained models inform a multiscale PN filtration model so fewer drastic assumptions are necessary compared to traditional pore-scale straining simulations. They also provide insight into which physical features of the pore space are most impactful on conductivity reduction due to straining. A common criticism of PN modeling is the lack of detailed geometry in simulation. By coupling with FEM, we attempt to make up some ground in this area, using more information to inform the hydraulic conductivity parameter. Although meshing challenges still exist, the local refinement capabilities of the unstructured approach make FEM especially favorable over other popular sub-pore-scale techniques such as LBM in simulating sub-pore-level fluid flow.

Once this framework is fully implemented, we will be able to compare the PN filtration model with and without FEM-coupling to experimental results from literature (specifically, the rates of permeability and porosity reduction and localized pressure drop in a glass bead pack). Khan, Mirabolghasemi et al. (2016) studied straining and deposition for particulate flows through glass bead packs using a combination of experiments and pore-scale simulations. They made some of the image data from that study available through the Digital Rocks data repository (Khan, Prodanovic et al. 2016), allowing us to directly compare our results to their models and experiments. More clean simulation data is required to make a proper multiscale filtration model for comparison purposes, but the tools and framework are now in place for such validation to occur in the near future.

4.2 Background

The process of physically removing particles from a fluid by way of flowing the particulate solution through porous media has been studied extensively, and can be referred to as particle size exclusion, interception, straining, or filtration. There are applications in geologic (petroleum, geothermal, aquifer) reservoirs (formation damage, fines migration, sand, suspensions), superabsorbent porous materials (hygiene products, construction materials), chemical processes (packed bed reactors, separations), soil science, and numerous other fields. The early studies in particle retention were mostly theoretical (Yao, Habibian et al. 1971) or experimental.
(Heertjes and Lerk 1967, Todd, Somerville et al. 1984, Vetter, Kandarpa et al. 1987), with a mixture of experimental and computational research in more recent years (Khan, Mirabolghasemi et al. 2016). There have even been pore-scale (Rege and Fogler 1988), Chen, Packman et al. (2008), Mirabolghasemi, Prodanović et al. (2015), Yang and Balhoff (2017)) and multiscale studies of solute transport (Mehmani and Balhoff 2015) and particle straining (French 2015), which are the central themes of this work. As far as I know, however, there have not been multiscale models between the pore and sub-pore scales, nor has this problem been approached through machine learning or data-driven modeling. In this work a hybrid, multi-model approach will be used to improve simulations of filtration phenomena (permeability reduction, penetration depth) and predict relevant properties (dispersion and retention coefficients).

Yang and Balhoff (2017) recently developed an advanced PN model for simulating particle retention. They included many mechanisms by which particles become immobilized, providing insight into which mechanisms dominate under various flow scenarios. While incorporating more phenomena increases the realism, one of the assumptions in their model was complete reduction of hydraulic conductivity in the case of size exclusion. Shi, Prodanović et al. (2013) coupled solid and fluid mechanics modeling of formation damage near wellbore. They provided two new models for throat conductance in a damaged formation. One was based on surface deposition, using an updated throat porosity ($\phi$) to determine a new equivalent radius ($R_{eq}$) which is then used in the Hagen-Poiseuille equation for calculating the conductivity ($g_{ij}$) of the throat connecting pores $i$ and $j$.

$$\pi R_{eq}^2 L = \pi R_{ij}^2 \phi, \quad R_e = R_{ij} \sqrt{\phi}, \quad g_{ij} = \frac{\pi R_{eq}^4}{8L} = \frac{\pi R_{ij}^4}{8L} \phi^2$$

Although it was developed for surface deposition, Khan, Mirabolghasemi et al. (2016) used this formula for calculating updated hydraulic conductivities of throats after entrapment of large particles. They used a CFD-DEM (Computational fluid dynamics – discrete element method) model to determine the locations where large particles became immobilized. The other model is based on clogging, and considers both tortuosity ($\tau$, assumed 2.5 for granular media) and particle diameter, $D_p$. However, their PN assumed cylindrical tubes, so particles larger than throats “were assumed to completely block the throat”, which I take to mean reducing the conductivity to 0 as Yang and Balhoff (2017) did. The clogging they refer to is a cluster of particles getting trapped in a throat and becoming its own granular medium within the throat, with its own porosity, $\phi$, and permeability, $k$.

$$g_{ij} = \frac{kA}{\mu L_{ij}}, \quad k = \frac{\phi^3 D_p^3}{72\tau(1 - \phi)^2}$$

In the current work, we focus on improving the assumptions which have traditionally gone into PN modeling of size exclusion. We believe it is not appropriate to broadly state that plugged throats are 100% blocked (that would require the particle geometry to be identical to the throat geometry, such as a perfectly spherical particle in a throat with perfectly circular cross-section. One can
imagine only a few physical scenarios where that would be the case, such as packings of synthesized hollow tubes). The porosity-based adjustments used in the recent works previously mentioned make another assumption which we challenge in this work – that throat conductivities will consistently be reduced more by interception of a large particle than a small particle. An additional challenge arises when the conductivity is reduced to a nonzero value – it is necessary to re-compute new geometric properties for the plugged pore and throat, which likely have nearly annular shapes.

The machine learning approach for transferring information between PN and FEM models was proven capable of determining reasonably accurate expressions for the hydraulic conductivity of throats. While improvements to the individual components of the multiscale model (especially meshing and PN extraction) would offer more confidence in the approach, an application of the model is presented here to illustrate the utility of the tools developed for this framework. We introduce a conductivity reduction parameter $\gamma_{ij}$ for updating the conductivity of the throat between pores $i$ and $j$, when a particle is intercepted. Larger values of this parameter indicate higher conductance, and therefore, less reduction of the original conductance. The conductivity of a throat with a trapped particle, $g_{ij}^{TP}$, is

$$g_{ij}^{TP} = \gamma_{ij} g_{ij}.$$

4.2.1 Converging-Diverging Elliptical Cross Section

Here we demonstrate the fact that larger particles do not necessarily cause more conductivity reduction than small particles in every throat, using a simplified example of particle straining in a single flow path with converging-diverging elliptical cross-section. An elliptical throat plugged with a spherical particle will have a nonzero conductivity, regardless of the size of the particle. Smaller particles will be trapped nearer to the tightest point in the constriction and will create more resistance than larger particles, which will be immobilized nearer to the pore center. A very simple, artificial, pore system was generated – a four-pore series of converging-diverging elliptical cross-sections. Figure 4.1 shows the five meshes which were used for testing the impact of particle size on throats of this geometry. A constant pressure drop was applied across the five domains, so that computing the flowrate would reveal the change in conductivity associated with increasing particle size. The results (Figure 4.2, Table 4.1) confirmed our hypothesis – that for such geometries, smaller particles actually have a more devastating impact on conductivity than larger ones. One can easily imagine a pore/throat geometry where the opposite trend is seen, but this example was shown to illustrate the less intuitive scenario. It should be noted that further increasing the particle size to the inscribed radius of the pore would have started impacting the previous throat in the flow path, leading to a lowering of the system’s total conductivity.
Figure 4.1. Series of converging-diverging elliptical ducts. Five meshes of increasing trapped particle size in a single throat. Top: external view of simple four-pore system. Bottom: slices through center of meshes to reveal higher element density near particle locations.

Figure 4.2. Flowrate through elliptical converging-diverging tube for various trapped particle sizes.
4.3 Methodology

4.3.1 Overview

Using machine learning to predict throat conductivity values is especially important for straining simulation, even for computing steady-state flows in the network prior to particle injection, because the flow field is dictated by phenomena at the pore-scale. Assuming a single geometry across all throats can result in a flow field that differs in both scale and distribution of flow, and therefore cause particles to follow different paths than they realistically would. This varying of distributions was shown in Chapter 3.

In generating data for throat conductivity prediction, subdomains of the pore space were meshed and coupled to the full domain PN model. FEM-computed conductivities of all meshed throats and their corresponding PN-derived features were used to train a machine learning model. To determine the effect of strained particles on a throat’s conductance, more data is required, including digitally-inserted artificial particles in a distribution of sizes, trapped in throats with a distribution of sizes/geometries. A code was developed to determine the approximate location where a particle of a given radius would become immobile if strained in any given throat. The algorithm uses the labeled voxel image and the network-identified pore and throat locations/sizes.

4.3.2 Adding trapped particles

When using the hybrid multiscale approach to predict throat hydraulic conductivity, generating large amounts of training data for the machine learning algorithms was straightforward – For a given material, once the mesh resolution necessary to consistently compute accurate throat flowrates was determined, it was just a matter of meshing as much of the material as was feasible (time and memory-wise), and doing this on several unique regions of the domain until enough data was generated. Every meshed throat

<table>
<thead>
<tr>
<th>Particle Radius (voxels)</th>
<th>Flowrate (vox³/s) @ ΔP=1</th>
<th>Reduction Parameter, $g^{TP}<em>{ij} / g</em>{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Particle</td>
<td>1.65</td>
<td>1.0</td>
</tr>
<tr>
<td>3.5</td>
<td>0.312</td>
<td>0.19</td>
</tr>
<tr>
<td>4</td>
<td>0.702</td>
<td>0.43</td>
</tr>
<tr>
<td>5</td>
<td>1.19</td>
<td>0.72</td>
</tr>
<tr>
<td>7</td>
<td>1.35</td>
<td>0.82</td>
</tr>
<tr>
<td>9</td>
<td>1.41</td>
<td>0.86</td>
</tr>
</tbody>
</table>
could yield another data point for training the model. To predict the reduction in conductivity of throats with particles trapped in them, a more tedious (and computationally expensive) approach is required.

For each data point in the training set, we need a finite element solution of flow through a throat with and without a particle inside it. Particles are typically much smaller than the grains which make up the porous material. Mesh resolution near particles must be fine in order to resolve the spherical surface, especially near contact points. There will be at least two contact points between any trapped particle and the material surface. These factors necessitate the use of localized mesh refinement. Even with an appropriate mesh with varied resolution near throats and particles, these meshes will incur greater computational costs than those without added particles. Therefore, only a limited number of particles can be included in a single mesh and still be computationally feasible.

Artificial particles must be added to throats in a manner that reflects the physical process of entrapment. For networks extracted based on skeletonization of the porespace, finding trapped particle locations would involve moving the sphere along the skeleton until it touched a solid surface. In this work, PNs are based on maximum inscribed spheres, so there is no skeleton to work with. For a computer-generated material, the location, size, and shape of every grain that composes it is known, making it straightforward to find the exact location where a spherical particle will be trapped in any given throat using geometry alone. For an image of a real porous material, we must rely on the information about the void-solid interface, determined during image segmentation.

Meshing trapped particles with varying particle and throat properties is difficult to do in a systematic way. Initial versions of the particle meshing scheme just used the location of the PN throat (defined as the position of the minimum inscribed sphere) regardless of the specified particle radius. In situations where the particle was larger than the throat, significant overlap of the two solid surfaces occurred. Since a reason for coupling the FEM and network models is to eliminate oversimplifications such as this, we want to make the resting position of the trapped particle physically realistic by minimizing the distance to the throat location without overlapping the solid phase. A particle placement scheme was developed which uses the PN, poremap, and burnmap. The burnmap is the voxel image labeled according to voxel burn (integer) distances to solid surfaces. A burn is just an outward (or inward, depending on the application) level-by-level voxel search. This can be used in the network extraction process for pore seed locations (initial estimates of where pores will be located). It is similarly used here to provide initial estimates of particle locations with the pore space. For a given throat, entry pore, and particle size, the process for determining final particle location is shown in Figure 4.3 and detailed below:

- Starting at the throat location (center of the yellow sphere in Figure 4.3 B), we burn outward, using the poremap to ensure we are within the pore from which the particle entered the throat.
The burn continues until a voxel with the same burnmap value as the particle radius is found (The next burn level is also searched to check if a closer (floating point number) location with that burnmap value exists. Two voxels at the same burn level are not necessarily the same distance from the throat location due to alignment with the coordinate direction). The voxel position encountered with the largest burn number which is less than the particle radius is also tracked, to bound the location. (Burnmap visualized in Figure 4.3 C)

- True distance calculations between the current location and the surrounding surfaces are used to refine the location of the trapped particle. Final coordinates are interpolated between the two bounding locations.

Figure 4.3. Visual depiction of particle placement process. A. Ball-and-Stick representation of PN superimposed on transparent poremap (only voxels associated with the two pores of interest). B. Purple and yellow spheres are the pore and throat inscribed
radii, respectively. C. Orthoslice of burnmap superimposed on poremap. D. Particle in its immobilized position. Neighbor pore made opaque for depth.

The particle placement algorithm has options specifically aimed at mesh generation for obtaining sufficiently large data sets. The throats which particles will be added to can be specified directly as an array of indices (which would be useful for calling this from another code), or by selecting one of the automated methods. These include (1) placing particles in all throats with no overlapping, (2) Randomly placing particles with a specified degree of separation (DOS). DOS is the minimum number of pores between any two particle-blocked throats. It was found that DOS should be at least 2 to avoid interference. The maximum number of mesh batches and minimum number of particles per mesh are also specified, and the algorithm continues to generate sets of particle location data until the criteria have been met. The particle size can be specified as a relative or absolute value. For absolute particle sizes, there fewer viable throats for particle placement because the inscribed pore and throat radii must be larger and smaller than the particle, respectively. Therefore, normalized particle size is better for data generation. Normalized particle sizes go from 0.0 to 1.0 for particles equal to the throat and pore inscribed radii, respectively.

4.3.3 Particle Meshing

A geometric meshing functionality, which was developed for meshing artificial media such as sphere packs and adopted for meshing nanoparticles adsorbed onto surfaces, is used in this work to add the strained particles into the meshed domain. The ability to use both geometric and voxel-based meshing in a single algorithm makes it possible to produce the necessary data without re-creating the voxel image with particles added as solid-phase voxels, in which case the resolution of the particle would be fixed to that of the voxel image. This is an important benefit given that the scale of the particle can potentially be much smaller than that of typical grains of the material, and would therefore require a higher resolution for accurately simulating the flow field in its vicinity. Another optional functionality in the in-house meshing workflow which is vital to modeling this type of hydrodynamic scenario is systematic mesh refinement at throat locations (constrictions, areas of increased local pressure gradient and velocity) and particle surfaces. Using manual refinement would be impractical in pore-scale modeling applications, where the number of meshed throats (and therefore regions of refinement) is often on the order of thousands or more, even for moderately sized subdomains. As in the previous chapter, the network model information is used to increase the distribution of points for mesh refinement at all throat constrictions. This ensures narrow throats are not closed off in mesh generation process, and improve flow solution accuracy and local mass balances. The largest pressure gradients occur inside throats, not larger pores. This is especially true when a foreign particle becomes lodged in the constriction, adding to the surface area and decreasing the cross-section, so increasing node density in these regions will provide more accurate solutions.
This application of the hybrid multiscale approach to particle filtration is one of the most illustrative examples of the advantages of FEM over other sub-pore scale methods. Although generating a proper, representative mesh is still quite challenging, the ability to create artificial particles at a resolution that is not tied to the image resolution, and to systematically refine the mesh around particle-throat contact points makes the generation of this sort of data possible. Along with the mesh refinement capabilities mentioned in 3.2.3.2.1, additional local refinement options have been developed (Farahani 2014) which create spherical shells at a resolution independent of the image and other meshing parameters. This makes it possible to add artificial spherical objects into the mesh which are not present in the voxel image. The scheme uses similar control options to those of the PN throat refinement procedure (location, refined resolution, buffer, decay exponent, and extent of refinement), and includes the sphere radius, which dictates the boundary of the shell. In order to produce well-behaved meshes, the transition between regions of very different resolution must be smooth. The extent and decay exponents should be carefully chosen to avoid very long, spoke-like tetrahedrons.

While it is always important to have a physically-representative mesh, it is imperative in the region of a particle trapped in a constriction between pores. There will be at least two points of contact between the spherical particle and the solid grains of the material, usually three. Farahani (2014) demonstrated the importance of contact point refinement in granular systems, and how coarse meshes tended to smooth the surface at the cusp where two round objects meet. That work dealt mostly with drag force calculations of colloidal particles, with radii up to 100 times smaller than the solid grains to which they were attached. Here we are interested in the increased resistance due to physical straining of particles on the order of the pore and throat inscribed radii. Therefore, while the same meshing challenges are present, their impacts on this work must be further quantified. Figure 4.4 shows how the contact point between an artificial particle and the solid interface of the image is handled by meshes with increasing refinement at the particle shell.
4.3.4 Computing Hydraulic Conductivity in Plugged Throats

The work presented in Chapter 3 provided a framework for improving pore network modeling capabilities. The reason for developing the iterative network pressure (INP) approach for computing throat conductances becomes clear when the hybrid modeling framework is applied to particle trapping simulations. When a relatively large particle is trapped in a pore, it physically divides the pore volume into several separate pore spaces (such as in Figure 4.5). In this case, the PLP method is not an option, since there is no element at the pore location and interpolating there will only yield nonsensical values. The MGP method would also likely fail due to multiple local minima within the (now divided) pore and none of them being likely to represent the whole pore.

Essentially, those two methods fail because the reality of the situation is that there is no longer a single pore, but several. If one must force the entire impact of the trapped particle onto one throat, the INP method provides that option.

Figure 4.4. Meshing near a contact point with varying levels of particle refinement. Each of these meshes were generated using the same initial point distribution prior to particle refinement, which reflected moderate surface and PN throat refinement. A. Particle shell exclusion with no additional refinement. B. Shell node spacing ~50% that of surface-refinement. C. Shell node spacing ~15% that of surface-refinement. D. Shell node spacing ~7% that of surface-refinement. Further increasing of resolution near contact points provides diminishing returns due to the expense of high resolution meshes and the relatively low flowrates in the immediate vicinity of the contact.
Figure 4.5. Trapped particle splitting a pore. Star indicates pore center location, which is contained by the trapped particle. A valid pore pressure cannot be interpolated from the location once the particle becomes trapped. In reality, the particle strongly impacts the conductivities of all throats connected to that pore by drastically altering the pressure field within the pore.

The same procedure is used in the presence of trapped particles, and the conductivity reduction parameter is computed by dividing the FEM-computed conductance of plugged throats by the results from before particles were added.

4.3.5 Pore Network Particle Filtration Model

French (2015) developed a pore network model for simulating transport and retention of mobilized particles in porous media. The model simulates filtration by size exclusion (Sharma and Yortsos, 1987a). While there are other particle retention mechanisms (diffusion, adsorption, gravitation, etc.) which play important roles in some cases, straining (aka interception or size exclusion) is the only retention mechanism considered by the model at this time. Size exclusion is known to be a dominant retention mechanism for micron-sized particles (Rege and Fogler, 1987). Rege and Fogler (1987) showed that permeability damage due to straining tends to be catastrophic, whereas traditional deep bed filtration models show gradual damage. The flow-biased particle path methodology used in that work is also used in the French (2015) PN filtration model, which operates as follows.

1. The number of injected particles, their size distribution, and injection times are model inputs.
2. An initial solution is computed using a traditional single-phase PN simulation. From this the steady-state pressure distribution, throat flowrates, total flowrate, initial effective porosity, and initial permeability are known for a given imposed pressure drop across the PN.
3. Particles are injected in series, and the probability of a particle flowing into a given throat is assumed to be stochastic, but proportional to the fluid flow rate in that throat. Therefore, particles are more likely to enter throats with higher flow rates.
Each particle’s injection location is also determined using a flow-biased probability calculation that considers the flow rates of all pores and throats on the injection face.

4. Each time a particle passes from one pore to another, its total transit time is updated by dividing the pore volume of the next pore by the volumetric flow rate between the pores. This provides an estimate of the time at which each particle was retained or exited the model.

5. If a particle enters a throat with an inscribed radius smaller than the particle radius, the particle is irreversibly retained. At this point, PN properties are recalculated for the plugged throat and pore (the pore from which the particle entered the plugged throat). The pore’s void volume is reduced by the volume of the retained particle. The cross-sectional area of the throat is reduced by the cross-sectional area through the center of the particle. The throat radius is updated by assuming the same ratio of inscribed to area-equivalent radii as before plugging.

6. Hydraulic conductivity of the throat is recomputed using updated properties and a traditional conductance approximation formula, unless the particle cross-section is larger than the throat’s cross-section. In this case, a 99% reduction in conductivity is assumed.

7. The location at which a particle is retained is recorded so that a spatial distribution of retained particle penetration can be calculated.

8. The PN simulation is run again to account for the change in flow field caused by the updated throat conductivity.

The conductivity calculation (steps 5 and 6) of the filtration model is where the multiscale model proposed in this work can provide improvement. The conductivity update outlined above essentially assumes the particle is located at the center of the throat’s inscribed radius, and that it overlaps the solid phase of the medium, although the particle’s volume is contained within only one pore. Some preliminary permeability reduction results using the PN filtration algorithm are shown in Figure 4.6. Here, we are assuming 99% reduction in hydraulic conductivity for blockages where the particle is larger than the throat’s equivalent radius and a reduction related to the reduction in cross-section otherwise. In either case, these assumptions oversimplify the model by not accounting for the actual location of the trapped particle. Two injection schemes are tested for three particle sizes injected sequentially. Starting with smallest particles, and successively increasing particle size every 0.75 pore volumes is shown at the top. In the bottom graph, we are starting with largest particles, and successively decreasing. Due to the nearly complete reduction in conductivity for plugged throats, once the inlet pores are all plugged with particles, the permeability quickly goes to 0. When large particles are injected first, this happens almost immediately. When small particles are injected first, some pathways remain
unplugged until the largest particles are finally injected. A more realistic adjustment to throat conductivities would show slower rate of permeability damage in all injection scenarios.

Figure 4.6. Permeability reduction due to particle straining. PN Filtration model result assuming 99% reduction in $g_{ij}$ for blockages where particle is larger than throat. Two injection schemes tested for 3 particle sizes injected in series – (top) Starting with smallest particles, successively increasing particle size every 0.75 pore volumes. (bottom) Starting with largest particles, successively decreasing.

4.4 Results and Discussion

4.4.1 Mesh Refinement/Dependency

To quantify the impact of particle mesh refinement on a throat’s conductivity reduction, a single particle in a single throat was analyzed. The mesh of the material with no particles added was generated first, using variable refinement near solid surfaces and at
throat locations (identified by the PN). The initial point distribution was used as the base distribution in several particle-refined meshes. This way, the meshes were nearly identical away from the particle, and the differences in solutions could be solely attributed to the particle refinement. Figure 4.7 shows the sensitivity of conductivity reduction to mesh refinement, and Figure 4.8 provides a visualization of several meshes corresponding to data points in Figure 4.7.

4.4.2 Hydraulic Conductivity of Blocked Throats

Using the glass bead pack image, a data set was generated using the particle placement algorithm and coupled FEM and PN models. Across 16 subdomains, approximately 500 throats were tested with 8 particle sizes throughout. The computed conductivities for throats with trapped particles were computed and divided by their conductivities prior to particle addition. The conductivity reduction parameters for all the data are shown in Figure 4.9, colored according to the normalized particle size, and plotted against the no-particle conductivities. Throats with very low initial conductivities were more likely to have above-average $\gamma_{ij}$. There seems to be a tendency for larger particles (relative to the pore size) to yield higher conductances (as in the elliptical converging-diverging example), but this is not universally the case.

A histogram of the entire data set is provided in Figure 4.10 to show the bimodal distribution of conductivity reduction parameter. The mean reduction parameter computed across all particle sizes was 0.225. Approximately 400 observations had reduced conductivity less than 1% of original conductivity, and 56 of those had a reduction parameter of 0 (meaning there was complete blockage of the throat to the scale of the mesh resolution).
Figure 4.7. (top) Blocked Throat Conductivity, $g_{ij}$, and (bottom) Conductivity Reduction Parameter, $\gamma_{ij}$, for various levels of mesh refinement. Red line designates the number of elements in the mesh before adding the particle.
Figure 4.8. Meshes with varying particle refinement, corresponding to three data points on Figure 4.7.
Figure 4.9. Scatterplot of conductivity reduction parameter with original conductivity of unblocked throats for 8 particle sizes.

Figure 4.10. Histogram of conductivity reduction parameter for Glass Bead Pack data
The factors influencing the reduction of conductivity by the presence of a trapped particle include the pore and throat properties, the particle properties, and relative properties between the particle and the pore/throat. Many of the same features used in conductivity modeling were used again in predicting conductivity reduction as these provide a representation of the pore/throat geometry. Particle properties we considered include the radius, volume, surface area.

Because multiple particle sizes were tested in each throat, care had to be taken to avoid extreme data leakage. Training and test sets were created by partitioning the data so that all data points associated with a particular throat were in the same set. By using a simple random split of the data or stratifying based on target value, the test and training sets would have nearly the same information (e.g. 6 particle sizes for throat $ij$ would be in the training set, so predicting the other 2 particle sizes would be a matter of interpolation. In this case, overfitting to the training data would not reveal itself and would result in falsely high prediction accuracy.).

The boosted tree model performed slightly better than assuming a constant reduction. The RMSE of reduction parameter predicted by the boosted tree was 0.151 on the held-out test set (bottom of Figure 4.11). Assuming a constant reduction equal to 0.0 and 0.225 (the mean value of the observations) resulted in RMSEs of 0.300 and 0.198, respectively. While this is not a particularly impressive result, it does show that the reduction in conductivity is (a) not constant, and (b) capable of being predicted by a data-driven model. The features which were important in making the predictions are shown in Table 4.2. The relative pore, throat, and particle radii along with the throat length proved to be among the most important features. The $ij$ Length Ratio also seemed to be a useful predictor. This feature is the ratio of pore-to-throat-to-pore distance to direct pore-to-pore distance. It was included as an effective tortuosity of the local 2-pore space.

Table 4.2. Contributions of features to the Boosted Tree

<table>
<thead>
<tr>
<th>Feature</th>
<th>Number of Splits</th>
<th>SS</th>
<th>Portion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rp/Rj</td>
<td>26</td>
<td>81.87885</td>
<td>0.4608</td>
</tr>
<tr>
<td>$ij$ Equivalent Radius/Length</td>
<td>31</td>
<td>19.83770</td>
<td>0.1116</td>
</tr>
<tr>
<td>Rij/Ri</td>
<td>10</td>
<td>18.19862</td>
<td>0.1024</td>
</tr>
<tr>
<td>$ij$ Length Ratio (i to $ij$ to $j$)/(i to $j$)</td>
<td>22</td>
<td>16.55288</td>
<td>0.0932</td>
</tr>
<tr>
<td>Surface Area Increase</td>
<td>12</td>
<td>14.15170</td>
<td>0.0796</td>
</tr>
<tr>
<td>$ij$ Wetted/Total Perimeter</td>
<td>11</td>
<td>9.29358</td>
<td>0.0523</td>
</tr>
<tr>
<td>Cross-Sectional Area Reduction</td>
<td>16</td>
<td>7.40427</td>
<td>0.0417</td>
</tr>
<tr>
<td>CSA Reduction Approximation</td>
<td>8</td>
<td>4.53264</td>
<td>0.0255</td>
</tr>
<tr>
<td>$ij$ Shape Factor</td>
<td>9</td>
<td>4.44130</td>
<td>0.0250</td>
</tr>
<tr>
<td>Rp/Ri</td>
<td>3</td>
<td>0.94128</td>
<td>0.0053</td>
</tr>
<tr>
<td>Particle Size (Normalized)</td>
<td>2</td>
<td>0.45463</td>
<td>0.0026</td>
</tr>
</tbody>
</table>
Every model in the PN modeling literature (to our knowledge) concerning plugging conductivity impairment caused by particulate flows assumes that when more volume is displaced, the conductivity is reduced further. Intuitively this does make sense – a larger particle creates more drag force and therefore more resistance to flow. However, in the context of straining in real porous materials, where geometries are highly irregular and widely varied, this assumption does not hold. This is, in part, because a single trapped
particle often impacts the conductivities of multiple throats, but we only observe the impact on one throat. Figure 4.12 shows the variation in conductivity of throats with a single trapped particle as the particle size is increased.

Figure 4.12. Variation in relationship between conductivity reduction and particle size.

One cause of the low data quality seen in the initial sets of particle simulations is the interference between nearby particles. After noticing a large MSE between flowrates in one of the subdomains resulting from the INP scheme, the data was explored to reveal the situation was caused by two particles invading the volume of a single pore. This spurred the inclusion of the DOS input to control particle placement. The effect was studied further using the test particle (A in Figure 4.14) from the mesh refinement dependency tests (Figure 4.7). The conductivity of that throat after being blocked by the particle was found to be 46.42 cm$^3$ when no other particles were present in the mesh. Adding a second particle into the throat of a neighboring pore (B in Figure 4.14), the conductivity was 44.85 cm$^3$ (3.4% change). Adding a particle into the throat of a pore that was two DOS (degrees of separation) away, (C in Figure 4.14), the conductivity was 46.26 cm$^3$ (0.3% change). A follow-up test using INP on domains with DOS of 1 and 2 (Figure 4.13) showed that using two degrees of separation helped significantly lower the error between the flowrates of the two models, but at the expense of generating less usable data for each mesh. These findings suggested that setting DOS to 2 would be sufficient, but the situation in Figure 4.15 demonstrates that particles can still interfere with one another even at DOS=2. The interior of a meshed subdomain with trapped particles is shown in Figure 4.15, the mesh surface triangles are colored according to the element pore labels. The two particles are nearer than expected for a DOS of 2.
Figure 4.13. DOS Impact on INP Convergence. Mean square error for iterations of the INP (Iterative Network Pressure) method for FEM-computed conductivity determination. Meshes compared with DOS=1 (33 trapped particles), DOS=2 (6 trapped particles)

We note that the data in Figure 4.13 could also have been misleading because by increasing the DOS, we caused some different particles to be meshed. Therefore, a particle-throat scenario which was causing problems at DOS=1 might not have been included in the DOS=2 mesh.
Figure 4.14. Single and 2-Particle meshes. Testing impact of DOS on conductivity reduction determination. Particle A is the test particle. Reduction due to A was computed in the absence of other particles (top), with a particle in a neighboring pore throat (center), and with a particle 2 DOS away (bottom).
Figure 4.15. Particles placed too close together, despite specified degrees of separation.

Figure 4.16 shows a real example of a throat where blockages of larger particles caused less conductivity damage than smaller ones. The narrow throat opens up into a wide pore, where trapped particles have low impact on flow resistance. The small constriction would be blocked almost completely by a smaller particle, but the medium-sized one pictured is trapped in a corner.
129

Figure 4.16. Situation where a trapped particle would not significantly decrease the conductivity, and would have no impact on the cross-sectional area or inscribed radius of the throat. It would add to the surface area and decrease the pore volume. The particle location is the closest point to the throat location where the distance to the surface is equal to the particle radius.

4.5 Conclusions

Recent advanced pore network models in literature have included many transport mechanisms for particle retention to study the effects of electrostatic and van der Waals forces, Brownian motion, gravity, adsorption/desorption, and interception. However, weakly-held assumptions concerning the impact of physical straining (size exclusion) on a throat’s conductivity are being made, so the relative importance of each mechanism can be misleading. By assuming a complete conductivity reduction for all blocked throats, portions of the porespace could become stagnant in the model, allowing the other mechanisms to dominate.

Although this study has not yet revealed an appropriate model for predicting conductivity reduction, we have uncovered some additional considerations for pore-level modeling of size exclusion. Most notably, that plugged throat conductivities do not go completely to 0 most of the time, and that the relationship between particle size and reduction cannot be stated independently of a throat’s geometry. This became obvious during the development of the particle placement algorithm, and was confirmed by the data. It was reiterated with a simplified example of particulate flow through a single converging-diverging elliptical cross-section.

Considering the extreme variation in 3-dimensional pore and throat shapes for real materials, the reduction is certainly not constant,
and preliminary results of this study suggest it is nonlinear (with respect to the PN parameters). Furthermore, it may not be possible to achieve a high degree of conductivity reduction prediction accuracy by applying all the resistance associated with a trapped particle to a single throat, which is done in all size exclusion PN models to the knowledge of the author, including this one. We can expect that to be the case in stereotypical pore and throat geometries, where throats are much smaller than the neighboring pores and the entire particle’s volume gets tucked into a single constriction. But when a trapped particle’s volume approaches that of the pore’s volume, it will effectively split the pore into several distinct pores, connected to each other by smaller throats with nonzero conductivity.

Mesh refinement testing revealed a small amount of dependence of conductivity on refinement near particle-grain contact points. Coarse meshes will give underestimates of porosity, surface area, and hydraulic conductance. However, due to diminishing returns, uninhibited refinement is not suggested for the simulation of many particles or for large-scale data generation in a multiscale framework. This analysis was admittedly a very narrow study, and more geometries and particle sizes should be included in a full quantification of the error associated with coarse contact point meshing.

In the context of generating conductivity data for machine learning, the process of artificially placing particles within pore throats did not take flow or pressure fields into account, but was purely geometrical. This was done to maximize the amount of data. Conductivity is a function of the local geometry only, so it was initially thought that the best approach would be to simulate as many non-overlapping particles at a time as possible. However, unexpected pore and throat geometries can result in a particle spanning several pores, and influencing the pressure and velocity fields within pores blocked by other particles. A simple test case and additional result data revealed that at least two degrees of separation should exist between two throats with blocked particles to minimize this occurrence.
This work focused on studying and simulating the interplay of flow phenomena at the multiple scales of porous media, and it challenged some of the assumptions commonly made in network modeling.

A fast, image-based algorithm for determining the minimum characteristic length scale of porous materials was developed. It is a pore-scale model used for determining adequate sample size for continuum-scale modeling, and can be used to gain insight into the presence of anisotropy and heterogeneity. It was applied to computations of permeability, formation factor, and effective porosity, but developed in an open manner for use in computing other properties of interest (residence time distribution, dispersion coefficient, etc.). PNs extracted from images of real rocks were analyzed and a dolomite image was found to be too small for determination of continuum-scale parameters, while a sandstone image was above the characteristic length scale. Although it was not mentioned in the initial publication of this work, another useful aspect of the algorithm is its ability to identify boundary/edge-effects. These can be caused by the PN extraction (where only portions of pores are shown in the 3d image, and therefore pores tend to have lower volume near boundaries), imaging (where there was additional image noise or interference due to the sample holder; this is usually eliminated by cropping), or sample preparation (the physical cutting process of the material can alter the structure of the pore network by detaching grains from the consolidated matrix and/or deforming portions of the interior pore space.). Boundary effects present themselves as spikes or drops in the largest one or two simulated subdomains (iterations on the graph), but are not present in the results presented here.

5.1 Recommendations and Future Work

There are many sensitivity analyses to perform in order to take full advantage of the methodology developed here. The large number of steps in the workflow and adjustable parameters at nearly every step made it impractical to do an exhaustive analysis of the tuning of the hybrid model.

Unsupervised learning should be looked into – all the machine learning discussed in this work has been supervised, meaning the learning algorithm had access to a set of target outcome values were provided to guide the learning process. In an unsupervised learning problem, data points are clustered or organized based solely on inputs, and no targets are provided. Even though the FEM solution provides target outcome variables, it would be interesting to perform such a clustering procedure on the input data to visualize the natural clusters of pore/throat types that exist. Then supervised regression could be performed on each cluster individually. Depending on the number of clusters that naturally arise, this may require substantially more data since each cluster would need an adequate set of training examples. Another benefit to this way of looking at the data is that outliers are often identified in the process. This is useful for an algorithm improvement standpoint, and more so for a deeper understanding of the
modeling variable space. I have briefly toyed around with a few unsupervised algorithms (TSNE, K-means) for curiosity’s sake, but do not claim to have adequate expertise to make use of these algorithms.

One can see how this framework can be used for other types of flow phenomena like non-Newtonian fluids, non-laminar flow, multiphase flow, reactive flow, adsorption/desorption, surface roughness, etc.). Many scenarios which previously required severe assumptions can now be accounted for in a hybrid network model, as long as there is a finite element implementation of the underlying physics. To move to transient processes, like reactive flow, a new FEM solver would have to be developed which was mass conservative. Since multiphase problems are time-dependent (versus single-phase Stokes flow, which is steady state), the errors associated with mass balance could compound in each timestep and lead to unreasonably large errors.

Updates to the in-house FEM meshing codes are currently in development. Once ready they will likely improve the ability to generate valuable data for machine learning, and do so more efficiently than the current meshing capabilities.

A mass-conservative sub-pore-scale model is necessary for using this framework confidently in broader applications. This can be achieved through post-processing recovery of mass balance (Coon, MacLachlan et al. 2010), another implementation of Stokes flow which explicitly enforces mass conservation element-wise, such as a discontinuous pressure solution, or using a different type of model, such as LBM.

Increasing the complexity of the pore network model would supply more features for learning new conductance prediction expressions. Using the [ore body – channel – pore body approach (where the throats contain volume) should be tested. This wouldn’t have to replace current network since both sets of network properties could be stored in the same PN file.

The logic used to reconcile connectivities between the models should be used during the meshing routine so that the mesh is created with matching connectivity and does not have to be processed further before using. If the planar regression for throats is implemented as mentioned above, meshing at interfaces should place points on the plane to provide a clean, refined interface for flow computation.

Additional engineered features to consider extracting from the PN were thought of after the current results were obtained. They are likely to help improve prediction accuracy, but have not yet been tested. For example, the pore coordination numbers are known, but the number of shared neighbors between them would likely be more useful. This information is more likely to reveal relationships for conductivity, especially in combination with the cross-sectional areas of throats connecting them to the shared neighbors and the ratio of wetted to total perimeter. There are surely many more useful geometric features which can be derived
from the information currently available. Using advanced nonparametric machine learning techniques can take advantage of such information more readily than, e.g., least squares.

As this framework is implanted, users should continue building a library/database of FEM throat solutions with corresponding PN parameters. Native throat parameters (features) should be sufficient as the engineered features can be computed later. An integer feature indicating porous media type should also be included. For example, 1=structured sphere pack; 2= unconsolidated granular packing (nearly spherical); 3=consolidated sandstone; 4=consolidated carbonite; 5=tight/shale; etc. There may eventually be enough training data that the FEM portion is unnecessary and users can just use the network with the coefficients learned from the master training set. Or even better, a subset of the master training set could be extracted with the same statistical distribution of features to build a model tailored to the network of interest.

5.1.1 Improvement to the Pore Network Model

The issue of multiple connections between two pores must be addressed. This situation is currently not accounted for in any special way. When two pores are connected by two different throats, there should be unique properties for each. Currently, the properties are merged into one. This can impact the ability to predict accurate conductivities since a single connection with similar features will have drastically different flow resistance, and will be even more problematic in other applications (especially particle filtration). This also presents a problem for PN-based mesh refinement, as discussed in section 3.2.3.2.1. Some effort must be devoted to adjusting the network structure to allow for this possibility, and modifying all existing network codes to handle the new structure.

Improvements should be made to the throat interface identification in PN extraction. Jagged interfaces result from under-merged (or over-seeded) PN extractions. They are not representative of true constrictions and cause errors in meshing. They also have uncharacteristic throat properties which creates noise in the data (if in the observation set) and/or error in prediction. Also, use of the shape factor in conductance calculations assumes that accurate estimations of perimeter and cross-sectional area are available. These are not trivial to obtain from voxelized images of real, unconsolidated media, especially when the interface is very irregular. Counting the shared voxel faces and edges between two pores is simple, but the orientations of throats are rarely aligned with the imaging axes and the staircase effect of the voxels yields drastic overestimates of area and perimeter. Skeleton-based PNs use the throat location medial axis to find the perimeter and area by dilation until there is a closed loop. These two issues suggest projecting the throat faces onto the plane of orientation should follow, but planar regression imposes a computational demand on the efficiency-focused network model. If implemented, the volume would no longer be identical in the labeled poremap and final PN.
A problem frequently encountered in this study has been with the extreme geometries of pores and throats occasionally output by the network generation code, Vox2Net. Non-physical geometries occasionally show up in the network, and present themselves in the poremap (pore-segmented voxel image) as

a. jagged spokes
b. throats with very little contact with the solid phase (low ratio of \(\frac{\text{Perimeter}_{\text{wetted}}}{\text{Perimeter}_{\text{total}}}\))

These artifacts impact the computation of several properties and have been the culprit behind many of the outliers which were analyzed in the numerous sets of data for this work. Planar regression has been applied to the Vox2Grains algorithm to refine the grain-grain contact interfaces. This can be used on pore throats just like grain contacts. It would be helpful to all network modeling codes since the pore and throat properties would be more realistic, but it would especially benefit this work because the properties of the pores and throats are used directly as features in the machine learning model.

5.1.2 Modular Meshing

Depending on available computational resources, it could be beneficial to concurrently couple the network filtration and FEM models. One approach to this type of coupling is suggested here:

- First, a large mesh of sufficient resolution to accurately compute \(g_{ij}\) is created for the entire domain. This mesh would be pore index-labeled, but no boundary conditions would be specified.
- For large images, solving Stokes flow in a mesh of this nature will likely be intractable. Therefore, a modular meshing technique should be used which extracts portions of the larger mesh for simulation. The subdomain mesh used would be determined by specifying the throats for which \(g_{ij}\) is being computed and the number of “buffer” pores which should be included to minimize boundary effects. Boundary conditions would be applied from network solutions, as mentioned in previous chapters.
  - By doing this, the total computational effort would be increased since the solution within some pores would be solved for multiple times. However, since each subdomain solution is computationally feasible and independent of the others, the problem is easily parallelized or simply split into batches on various nodes.
  - Also, the machine learning approach can be used to predict remaining throats once a sufficient number of \(g_{ij}^{\text{FEM}}\) have been computed.
- Each time a particle becomes trapped in a throat, an FEM simulation of that local pore space, combined with the updated network connectivity mentioned in 5.1.1, would give updated \(g_{ij}\) values for the new throats. Once the subdomain mesh was
extracted, the particle location would be determined and the mesh would be adapted to include the additional trapped particle.

5.1.3 Improvements to the Pore Network Filtration Model
For the case of particle straining, where the particle size is not many orders of magnitude smaller than the pore/throat sizes, I propose a functionality to be incorporated into the filtration algorithm which modifies the PN model to account for the change in pore space geometry and interconnectivity. Pores containing large trapped particles should be split into multiple new pores, treating the particle as a portion of the solid phase. We should:

1. Use the Particle Placement routine described in 4.3.2 to find the location where the particle will finally become immobile.
2. Insert the particle by either
   i. adding solid voxels for particle (which would regrettably tie the particle resolution to image resolution, but would simplify the remaining steps.)
   ii. using a hybrid of the geometric-based and voxel-based network generation routines to discretize the pore, while keeping the particle resolution decoupled from the image (in which case the pore volumes in the poremap would not be the same as volumes in the PN. This may be acceptable, but is something to keep in mind).
3. Create a subdomain image around the pores and throats in the neighborhood of the domain, only including void voxels of pores which have had at least one of their voxels replaced by solid phase due to the new particle and are completely contained within the subdomain
4. Run Vox2Net on the subdomain image to get the new pores/throats. Use the same Vox2Net criteria used in initial network generation to maintain consistency.
5. Update imatP and do bookkeeping to insert new pores/throats with updated properties and interconnectivities into the PSN
Figure 5.1. Schematic of proposed network modification for size-exclusion particle trapping. Circles are solid phase, lines are network throat interfaces. Grey circles represent particles during trapping procedure, which become part of the solid matrix.

The computational demand for this would likely be quite high. It is not yet clear whether we can afford to do this on every plugged throat in real problems of interest. It would depend on:

- How many particles (max) are we wanting to add into network models?
- How efficient can we make the subdomain image extraction + subVox2Net?
- Time would likely be the limiting factor more so than memory. Though keeping the poremap and burnmap in memory will have an impact, I don’t think our network models have come close to the upper limit of memory (1TB node on SuperMike-II)

The main benefit to doing this will be for consecutive particles entering a single pore, especially in tight porous media where there are a limited number of flow paths through the domain and many particles are likely to get trapped in the same pores. By continuously updating the network interconnectivity, there is no need for making assumptions about when a pore becomes filled with particles (such as using volume and porosity, and assuming they will arrange nicely). The multiscale nature of the problem will be automatically handled, and we could infiltrate smaller and smaller particles and continue the process. By doing so we are creating smaller and smaller pores and throats, and the structure at that level becomes quite different (random sphere-pack) than the original porous medium (carbonate/sandstone). Therefore, it is important to adjust the prediction of throat conductivity in newly created throats between trapped particles to account for the new geometry, unless the conductivity prediction expression is proven to be robust to these differences. In the current work, we used FEM to predict $g_{ij}$ of plugged throats. This $g_{ij}$ of a plugged throat is really the series/parallel resistance of the new subdomain network within the blocked pore, considering the particle. So if we've
learned a method for predicting $\gamma_{ij}$, we can use it to ensure our series/parallel resistance of the new throats is equivalent to the predicted value.
REFERENCES


APPENDIX A. ACS RIGHTS LINK REUSE REQUEST AND PERMISSION

<table>
<thead>
<tr>
<th>Title:</th>
<th>Rapid Estimation of Essential Porous Media Properties Using Image-Based Pore-Scale Network Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author:</td>
<td>Timothy W. Thibodeaux, Qiang Sheng, Karsten E. Thompson</td>
</tr>
<tr>
<td>Publication:</td>
<td>Industrial &amp; Engineering Chemistry Research</td>
</tr>
<tr>
<td>Publisher:</td>
<td>American Chemical Society</td>
</tr>
<tr>
<td>Date:</td>
<td>Apr 1, 2015</td>
</tr>
<tr>
<td>Copyright:</td>
<td>© 2015, American Chemical Society</td>
</tr>
</tbody>
</table>

Quick Price Estimate

Permission for this particular request is granted for print and electronic formats, and translations, at no charge. Figures and tables may be modified. Appropriate credit should be given. Please print this page for your records and provide a copy to your publisher. Requests for up to 4 figures require only this record. Five or more figures will generate a printout of additional terms and conditions. Appropriate credit should read: "Reprinted with permission from {COMPLETE REFERENCE CITATION}. Copyright {YEAR} American Chemical Society." Insert appropriate information in place of the capitalized words.

<table>
<thead>
<tr>
<th>I would like to...</th>
<th>reuse in a Thesis/Dissertation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requestor Type</td>
<td>Author (original work)</td>
</tr>
<tr>
<td>Portion</td>
<td>Full article</td>
</tr>
<tr>
<td>Format</td>
<td>Electronic</td>
</tr>
<tr>
<td>Will you be translating?</td>
<td>No</td>
</tr>
<tr>
<td>Select your currency</td>
<td>USD - $</td>
</tr>
<tr>
<td>Quick Price</td>
<td>Click Quick Price</td>
</tr>
</tbody>
</table>

This service provides permission for reuse only. If you do not have a copy of the article you are using, you may copy and paste the content and reuse according to the terms of your agreement. Please be advised that obtaining the content you license is a separate transaction not involving Rightslink.
PERMISSION/LICENSE IS GRANTED FOR YOUR ORDER AT NO CHARGE

This type of permission/license, instead of the standard Terms & Conditions, is sent to you because no fee is being charged for your order. Please note the following:

- Permission is granted for your request in both print and electronic formats, and translations.
- If figures and/or tables were requested, they may be adapted or used in part.
- Please print this page for your records and send a copy of it to your publisher/graduate school.
- Appropriate credit for the requested material should be given as follows: "Reprinted (adapted) with permission from (COMPLETE REFERENCE CITATION). Copyright (YEAR) American Chemical Society." Insert appropriate information in place of the capitalized words.
- One-time permission is granted only for the use specified in your request. No additional uses are granted (such as derivative works or other editions). For any other uses, please submit a new request.
APPENDIX B. MASS BALANCE IN FEM SOLUTION

When the idea of coupling the FEM model with the network model was initially approached, it was expected that a network model could be made to agree exactly with an FEM model solution if both were representations of an entire sample domain. The premise was: if the FEM solution was used to compute the flowrate through each throat in the sample, a hydraulic conductivity for each corresponding throat in the network could be back-calculated from that result, and when the network was solved using that set of conductivities, the two models would be in agreement for both the overall flowrate and the individual throat flowrates. On the contrary, when this process was attempted, there were drastic differences between the results of the two models.

First, there were different numbers of throats in each model. This is due to the fact that the mesh and network generation processes are largely independent of each other and both have many “knobs” (adjustable parameters) which can considerably affect the numerical size of the discretization and how well the voxelized pore space is represented. (See Lane (2011) dissertation Chapter 5. The Effect of Image-based Meshing on Predicted Permeability, and Thompson (Al-Raoush, Thompson et al. 2003) paper on network model generation.) This problem was addressed on two fronts:

1. The minimum number of voxel-voxel contacts between two pores required to constitute a network throat was increased in order to avoid single voxel throats which are difficult to mesh properly without extreme local mesh refinement. This issue was discussed in 3.3.2.4 and (Thompson, Willson et al. 2008).

2. Once the network was generated, the locations of the throats were used as regions of interest for local mesh refinement. The mesh resolution was increased near the throats in proportion to the inscribed radius of the throat. (i.e. smaller spacing between initial point locations was used in smaller throats)

Despite these measures, discrepancies between the pore-to-pore connectivities of the two models persisted, though with far lower frequency. Since it is undesirable to continue increasing FEM mesh resolution and regenerating the network model from the voxel image due to time and computational effort, the option to force the two models into agreement was developed to be implemented as a last resort (a code adds or removes small, insignificant throats to the network in order to match the mesh connectivity, or relabels elements in the mesh to agree with the network).

After those discrepancies were fixed and the models were in agreement (connectivity-wise), the two sets of results still varied. Upon closer examination of individual throat flowrates, it was apparent that for some pores there was a net accumulation or consumption. Although the overall FEM Stokes model is mass conservative, when the fluxes through all throats connected to a pore were added up, the balance was not zero. In some cases the error in balance was significant relative to the total flow through the pore. Further
studies were performed to be sure there was not an error in the flow or velocity calculations. Structured meshes (explicit voxel mesh. See Lane (Lane 2011)) of a cylinder and an idealized two-pore, one-throat 3d image were used to investigate the mass balance error that should be expected. All the elements in these meshes have excellent condition number (Freitag and Knupp 2002), condition number as a tetrahedral mesh quality metric, 2002), and the faces on boundaries where Neumann boundary conditions are applied are all aligned with the Cartesian coordinate axes so no error is associated with element quality or with applying pressure as traction BCs (Lane 2011) demonstrated that the effect of the traction BC is relatively small). By summing up the flux through each of the 4 faces of every tetrahedral element, then dividing by the sum of positive fluxes, the elemental mass balance errors were obtained for analysis. The range of error percentages varied greatly, from 0% to 400+, but the elements with the worst errors all seemed to be near the solid-void interface, where the no-slip boundary conditions are applied. Because the structured mesh follows the “staircase” structure of the original voxel data, many elements end up in corners. Although an element situated perfectly in a corner presents no problem (because all of the nodes would have Dirichlet no-slip BCs applied on them and the flow through every face is exactly 0), when there is a single node on an element that has a non-zero velocity, the error intrinsic to the flowrate calculation becomes apparent. See the sample calculation below for additional details.

It was determined that even if the velocity solution is certainly correct (i.e. the same as an analytical solution) at every point in the mesh, there can still be mass balance errors when integrating over element faces due to the fact that local mass conservation is not explicit for the traditional Taylor-Hood element (P₂P₁) with the Galerkin FEM (GFEM), which is used exclusively in this study. It turns out this is a well-known issue, despite being unknown to the author prior to this study. Gresho and Sani (Gresho and Sani 2000) pointed out that although discontinuous pressure elements do not possess uniquely defined pressure on element boundaries (they are dual-valued there, and often multi-valued at certain velocity nodes), only discontinuous pressure elements assure an element-level mass balance. They also gave this as proof: for \( \psi_i = \text{piecewise constant on element } e \),

\[
0 = \int \psi_i \nabla \cdot \mathbf{u}^h = \int_{\Omega_e} \nabla \cdot \mathbf{u}^h = \int_{\Gamma_e} \mathbf{n} \cdot \mathbf{u}^h
\]

and only discontinuous pressure elements contain this element-level test function. So, the continuous pressure approximation of the P₂P₁ element cannot deliver element-level mass balances. Discontinuous pressures sidestep each of these disadvantages. Using a discontinuous pressure approximation in the discretization of the Stokes problem, such as P₁P₀ elements, the mass conservation would be satisfied more locally, but the accuracy of the solution and convergence stability will suffer. Many researchers have circumvented this problem by developing alternative FEM approaches such as discontinuous, hybrid, and mixed formulations. These methodologies enforce local mass conservation, but at a cost. Others assert that there is a trade-off between the Taylor-Hood
element (better accuracy, cheaper computation, local conservation not guaranteed) and discontinuous approximations (often less accurate and less stable, difficult implementation, expensive computation, guaranteed local conservation). For the current study, the $P_2P_1$ Taylor-Hood element was used despite the lack of guaranteed local conservation due to the benefits mentioned plus the ease of implementation. Given more time to devote to this project, a next step for overcoming this issue would be to implement a $P_2(P_1 + P_0)$ element such as Tidd et al. (Tidd, Thatcher et al. 1988), or to apply some post-processing scheme for conservation enforcement such as Hughes et al. (Hughes, Engel et al. 2000) Other researchers with similar goals have also developed post-processing algorithms to attempt improving the local conservation of the Galerkin method (Cockburn, Gopalakrishnan et al. 2007, Coon, MacLachlan et al. 2010), and have been successful in their efforts, so it would be important to consider each of these approaches. These will be discussed further in the Future Work section of this document. For the current work, the author decided to press on, despite knowledge of the inherent flaw in the current implementation, and hopes the reader will be convinced by the following data that the errors attributed to mass balance are relatively small, and sufficiently accurate throat flowrates can be computed using the traditional CGM.

The following is an explanation and sample calculation of the flow through a face of a specific corner element which had the no-slip Dirichlet BC applied to nine of the ten nodes. Two of the four faces were in contact with the solid phase, and therefore had no flow through them. The other two faces were open to flow, and shared the common non-zero velocity node.

The flowrate through a single face of a tetrahedral element is computed using 4-point Gaussian quadrature to integrate the normal component of the velocity solution over the triangular area of the face.

\[
\int_{-1}^{1} f(x)dx = \sum_{i=1}^{4} w_i f(x_i)
\]

The quadrature rules computed by Dunavant (1985) were used for the integration.

\[
QP \ Face \ (QP_1) = QP \ Weight_{triangle} \ * \ Area
\]

In Figure B.1, the element outlined in yellow contains only one node with nonzero velocity. Two of the four element faces are on the solid surface and will have no flux, so the other two should be equal but opposite nonzero flowrates. Sample flow calculations for the two faces are provided in Table B.1 and Table B.2.
Figure B.1. Close-up view of Taylor-hood element (outlined in yellow) in a corner of a voxelized surface. Taken from a structured voxel-based mesh where each voxel is split into 5 elements. The element has only one node (green) which is not no-slip.

\[ \text{Face 1 Normal} = [0, 1, 0] \]

\[ \text{Face 1 Area} = 0.5 \]

Table B.1. Quadrature point values and calculation for Face 1 flow rate.

<table>
<thead>
<tr>
<th></th>
<th>QP Triangle Weight</th>
<th>QP Face</th>
<th>QP Flow</th>
<th>QP Weighted Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP_1</td>
<td>-0.5625</td>
<td>-0.281</td>
<td>0.20</td>
<td>-0.056</td>
</tr>
<tr>
<td>QP_2</td>
<td>0.520833</td>
<td>0.260</td>
<td>0.21</td>
<td>0.0546</td>
</tr>
<tr>
<td>QP_3</td>
<td>0.520833</td>
<td>0.260</td>
<td>0.21</td>
<td>0.0546</td>
</tr>
<tr>
<td>QP_4</td>
<td>0.520833</td>
<td>0.260</td>
<td>0.07</td>
<td>0.0182</td>
</tr>
</tbody>
</table>

**Face 1 Flow (cm^3/s)** | 0.0746

\[ \text{Face 2 Normal} = [-0.577, -0.577, -0.577] \]

\[ \text{Face 2 Area} = 0.866 \]

Table B.2. Quadrature point values and calculation for Face 2 flow rate.

<table>
<thead>
<tr>
<th></th>
<th>QP Triangle Weight</th>
<th>QP Face</th>
<th>QP Flow</th>
<th>QP Weighted Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP_1</td>
<td>-0.5625</td>
<td>-0.49</td>
<td>-0.19</td>
<td>0.0931</td>
</tr>
<tr>
<td>QP_2</td>
<td>0.520833</td>
<td>0.45</td>
<td>-0.21</td>
<td>-0.095</td>
</tr>
<tr>
<td>QP_3</td>
<td>0.520833</td>
<td>0.45</td>
<td>-0.21</td>
<td>-0.095</td>
</tr>
<tr>
<td>QP_4</td>
<td>0.520833</td>
<td>0.45</td>
<td>-0.07</td>
<td>-0.032</td>
</tr>
</tbody>
</table>

**Face 2 Flow (cm^3/s)** | -0.1274

This element has a net accumulation of -0.0489, -65.5% of the flow (0.0746 cm^3/s) exiting the element at face 1. The full data for this calculation is shown in the top half of Table B.3.
Table B.3. Gaussian Quadrature calculation data for the element described in the sample flowrate calculation with large mass balance error. The second is an element with negligible error. Calculations are only shown for faces with significant flow (i.e. no-slip faces are not calculated).

<table>
<thead>
<tr>
<th>Element A</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Face 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Face area</td>
<td>0.5</td>
<td>0.8660</td>
</tr>
<tr>
<td>Face normal</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>QP Flow</td>
<td>0.20</td>
<td>0.21</td>
</tr>
<tr>
<td>28%</td>
<td>31%</td>
<td>31%</td>
</tr>
<tr>
<td>QP Wt</td>
<td>-0.28</td>
<td>0.26</td>
</tr>
<tr>
<td>-0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>Flowrate</td>
<td>0.0746</td>
<td>-0.1274</td>
</tr>
<tr>
<td>Mass Balance</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>positive</td>
<td>0.0746</td>
<td>-65.50%</td>
</tr>
<tr>
<td>negative</td>
<td>-0.1234</td>
<td></td>
</tr>
<tr>
<td>balance</td>
<td>-0.0489</td>
<td></td>
</tr>
</tbody>
</table>

Table B.4. Gaussian Quadrature calculation data for an element with negligible error. Calculations are only shown for faces with significant flow (i.e. no-slip faces are not calculated).

<table>
<thead>
<tr>
<th>Element B</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Face 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Face area</td>
<td>0.5</td>
<td>0.8660</td>
</tr>
<tr>
<td>Face normal</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QP Flow</td>
<td>-0.20</td>
<td>-0.22</td>
</tr>
<tr>
<td>28%</td>
<td>31%</td>
<td>10%</td>
</tr>
<tr>
<td>QP Wt</td>
<td>-0.28</td>
<td>0.26</td>
</tr>
<tr>
<td>-0.0760</td>
<td>0.0760</td>
<td></td>
</tr>
<tr>
<td>Mass Balance</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>positive</td>
<td>0.076</td>
<td>0.00%</td>
</tr>
<tr>
<td>negative</td>
<td>-0.076</td>
<td></td>
</tr>
<tr>
<td>balance</td>
<td>-4.0e-11</td>
<td></td>
</tr>
</tbody>
</table>


**APPENDIX C. ADDITIONAL DATA**

Table C.1. Glass Beads training data quantiles

<table>
<thead>
<tr>
<th>Quantiles</th>
<th>ij Inscribed Radius</th>
<th>i Surface Area</th>
<th>ij Cross-sectional Area</th>
<th>ij Inverse Throat Length</th>
<th>ij Perimeter</th>
<th>ij Wetted Perimeter</th>
<th>ij Shape Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0%</td>
<td>Max</td>
<td>23.78</td>
<td>9540.71</td>
<td>2933.28</td>
<td>0.447</td>
<td>688.00</td>
<td>604.00</td>
</tr>
<tr>
<td>99.5%</td>
<td></td>
<td>18.29</td>
<td>5410.38</td>
<td>1973.82</td>
<td>0.178</td>
<td>512.00</td>
<td>483.20</td>
</tr>
<tr>
<td>97.5%</td>
<td></td>
<td>15.59</td>
<td>3338.49</td>
<td>1264.00</td>
<td>0.088</td>
<td>374.00</td>
<td>326.50</td>
</tr>
<tr>
<td>90.0%</td>
<td></td>
<td>13.05</td>
<td>2337.57</td>
<td>799.24</td>
<td>0.057</td>
<td>268.00</td>
<td>231.00</td>
</tr>
<tr>
<td>75.0%</td>
<td>Quart.</td>
<td>10.36</td>
<td>1760.76</td>
<td>531.08</td>
<td>0.041</td>
<td>210.00</td>
<td>182.00</td>
</tr>
<tr>
<td>50.0%</td>
<td>Median</td>
<td>7.83</td>
<td>965.55</td>
<td>310.27</td>
<td>0.028</td>
<td>152.00</td>
<td>130.00</td>
</tr>
<tr>
<td>25.0%</td>
<td>Quart.</td>
<td>5.88</td>
<td>304.07</td>
<td>42.81</td>
<td>0.020</td>
<td>48.00</td>
<td>20.00</td>
</tr>
<tr>
<td>10.0%</td>
<td></td>
<td>2.75</td>
<td>133.37</td>
<td>12.77</td>
<td>0.016</td>
<td>24.00</td>
<td>8.00</td>
</tr>
<tr>
<td>2.5%</td>
<td></td>
<td>1.60</td>
<td>32.03</td>
<td>5.00</td>
<td>0.012</td>
<td>14.00</td>
<td>2.00</td>
</tr>
<tr>
<td>0.5%</td>
<td></td>
<td>1.48</td>
<td>11.78</td>
<td>3.16</td>
<td>0.011</td>
<td>10.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.0%</td>
<td>Min</td>
<td>1.00</td>
<td>3.78</td>
<td>1.41</td>
<td>0.008</td>
<td>6.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantiles</th>
<th>Wetted to Total Perimeter</th>
<th>Pore i CN</th>
<th>Pore j CN</th>
<th>i Inscribed Radius</th>
<th>j Inscribed Radius</th>
<th>i Volume</th>
<th>j Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0%</td>
<td>Max</td>
<td>1</td>
<td>33</td>
<td>21</td>
<td>27.84</td>
<td>22.72</td>
<td>558571</td>
</tr>
<tr>
<td>99.5%</td>
<td></td>
<td>1</td>
<td>23</td>
<td>13</td>
<td>26.19</td>
<td>19.65</td>
<td>508963</td>
</tr>
<tr>
<td>97.5%</td>
<td></td>
<td>1</td>
<td>18</td>
<td>10</td>
<td>23.81</td>
<td>17.52</td>
<td>283126</td>
</tr>
<tr>
<td>90.0%</td>
<td></td>
<td>0.98</td>
<td>13</td>
<td>7</td>
<td>20.46</td>
<td>14.96</td>
<td>158174</td>
</tr>
<tr>
<td>75.0%</td>
<td>Quart.</td>
<td>0.91</td>
<td>10</td>
<td>5</td>
<td>17.90</td>
<td>12.95</td>
<td>86207</td>
</tr>
<tr>
<td>50.0%</td>
<td>Median</td>
<td>0.79</td>
<td>7</td>
<td>3</td>
<td>14.84</td>
<td>10.29</td>
<td>37810</td>
</tr>
<tr>
<td>25.0%</td>
<td>Quart.</td>
<td>0.5</td>
<td>5</td>
<td>2</td>
<td>12.55</td>
<td>7.52</td>
<td>24796</td>
</tr>
<tr>
<td>10.0%</td>
<td></td>
<td>0.25</td>
<td>4</td>
<td>1</td>
<td>10.88</td>
<td>4.61</td>
<td>18159</td>
</tr>
<tr>
<td>2.5%</td>
<td></td>
<td>0.1</td>
<td>2</td>
<td>1</td>
<td>8.19</td>
<td>2.08</td>
<td>1488.5</td>
</tr>
<tr>
<td>0.5%</td>
<td></td>
<td>0.04</td>
<td>2</td>
<td>1</td>
<td>2.91</td>
<td>1.21</td>
<td>108.4</td>
</tr>
<tr>
<td>0.0%</td>
<td>Min</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1.27</td>
<td>1.21</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantiles</th>
<th>i Surface Area</th>
<th>j Surface Area</th>
<th>Inverse Length (i to j to j)</th>
<th>Rij/Ri</th>
<th>Equivalent Radius / Length</th>
<th>Hagen-Poiseuille Prediction</th>
<th>Actual gij</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0%</td>
<td>Max</td>
<td>47267.1</td>
<td>27109.6</td>
<td>0.40</td>
<td>1.17</td>
<td>1.36</td>
<td>3549.83</td>
</tr>
<tr>
<td>99.5%</td>
<td></td>
<td>38415.2</td>
<td>15744.8</td>
<td>0.16</td>
<td>0.99</td>
<td>1.12</td>
<td>1422.79</td>
</tr>
<tr>
<td>97.5%</td>
<td></td>
<td>23904.8</td>
<td>9125.0</td>
<td>0.09</td>
<td>0.93</td>
<td>0.93</td>
<td>762.20</td>
</tr>
<tr>
<td>90.0%</td>
<td></td>
<td>16439.7</td>
<td>5486.1</td>
<td>0.06</td>
<td>0.84</td>
<td>0.66</td>
<td>366.25</td>
</tr>
<tr>
<td>75.0%</td>
<td>Quart.</td>
<td>9398.9</td>
<td>3862.4</td>
<td>0.04</td>
<td>0.72</td>
<td>0.43</td>
<td>150.18</td>
</tr>
<tr>
<td>50.0%</td>
<td>Median</td>
<td>5012.1</td>
<td>478.1</td>
<td>0.03</td>
<td>0.56</td>
<td>0.24</td>
<td>44.41</td>
</tr>
<tr>
<td>25.0%</td>
<td>Quart.</td>
<td>3551.0</td>
<td>144.4</td>
<td>0.02</td>
<td>0.37</td>
<td>0.11</td>
<td>10.95</td>
</tr>
<tr>
<td>10.0%</td>
<td></td>
<td>2738.7</td>
<td>37.9</td>
<td>0.02</td>
<td>0.20</td>
<td>0.05</td>
<td>0.56</td>
</tr>
<tr>
<td>2.5%</td>
<td></td>
<td>267.9</td>
<td>18.2</td>
<td>0.012</td>
<td>0.11</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td>0.5%</td>
<td></td>
<td>34.6</td>
<td>5.3</td>
<td>0.010</td>
<td>0.08</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>0.0%</td>
<td>Min</td>
<td>12.2</td>
<td>0.566</td>
<td>0.008</td>
<td>0.047</td>
<td>0.010</td>
<td>0.004</td>
</tr>
</tbody>
</table>
Table C.2. Random sphere pack mesh refinement tuning parameters.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>R1</th>
<th>R2</th>
<th>Extent</th>
<th>Throat Refinement R3 (x R_i)</th>
<th>points</th>
<th>elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>10</td>
<td>20</td>
<td>0.75</td>
<td>9282</td>
<td>40414</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>8</td>
<td>16</td>
<td>0.75</td>
<td>16801</td>
<td>67236</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>6</td>
<td>12</td>
<td>0.75</td>
<td>17898</td>
<td>75273</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>4</td>
<td>8</td>
<td>0.75</td>
<td>44434</td>
<td>78482</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>18525</td>
<td>103380</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>5</td>
<td>10</td>
<td>0.5 **</td>
<td>28130</td>
<td>147022</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>4.4</td>
<td>9</td>
<td>0.4 **</td>
<td>46122</td>
<td>197776</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>0.75</td>
<td>46272</td>
<td>228561</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>0.75</td>
<td>68038</td>
<td>273168</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>4.5</td>
<td>9</td>
<td>0.75</td>
<td>108415</td>
<td>324668</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>0.75</td>
<td>166956</td>
<td>438074</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>3</td>
<td>9</td>
<td>0.75</td>
<td>346662</td>
<td>858346</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>0.75</td>
<td>778658</td>
<td>1795344</td>
</tr>
</tbody>
</table>

Constant meshing parameters for each:
- Buffer sizes (0.2 at surfaces, 0.0 at throats)
- Decay Exponents (1.5 – surfaces; 1.0 – throats unless marked ** 0.5)
- Throat refinement extent = pore inscribed radius
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description/Notes</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Fatt 1956) (Salter and Mohanty 1982, Lenormand, Touboul et al. 1988, Vrettos, Imakoma et al. 1989, Blunt and King 1990, Imdak and Sahimi 1991, Mani and Mohanty 1999, Joekar-Niasar, Hassanizadeh et al. 2008)</td>
<td>Hagen-Poiseuille flow through a circular tube. $R_t$ was simply the tube radius in early models. There are many variations on this formula throughout network modeling literature, with various methods of calculating radius and length.</td>
<td>$g_{ij} = \frac{\pi R_t^4}{8 L_t}$</td>
</tr>
<tr>
<td>(Cerny and Walawender 1966)</td>
<td>Flow of viscous liquid in a converging tube</td>
<td>$g_{ij} = \frac{3\pi a^4}{4(2 - 3 \sin^2 \alpha) (\cos^3 \alpha) \left(\frac{1}{l_1^2} - \frac{1}{l_2^2}\right)}$</td>
</tr>
<tr>
<td>(Nonweiler 1975, Lewis and Boose 1995)</td>
<td>Author use correction factor to account for the conduit shape and eccentricity. $k=2$ or $e=0$ (circular cross section) $k=4$ or $e=1$ (completely flattened ellipses) $k=2.03$ (ellipses with $e=0.5$) $k=1.78$ (square) $k=3$ (narrow slit)</td>
<td>$g_{ij} = A_i \frac{D_p^2}{16kL_T}$ $k = \frac{1 + \sqrt{(1 - e^4)}}{\sqrt{b^2 - a^2}}$ $e = \frac{b}{a}$ $a = R_{inscribed}$ $b = \sqrt{\left(\frac{P}{\pi}\right)^2 - R_{inscribed}^2}$</td>
</tr>
<tr>
<td>(Payatakes, Ng et al. 1980, Lin and Slattery 1982, Dias and Payatakes 1986, Hopkins and Ng 1986)</td>
<td>The parameter $c_2$ is a constant specific to a porous medium. $c_2=2.652$ used for a glass bead pack, 5.25 for a sand packing.</td>
<td>$g_{ij} = \frac{2\pi c_2 R_t^3}{-\Delta P_1}$</td>
</tr>
<tr>
<td>Reference</td>
<td>Description/Notes</td>
<td>Equation</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>(Pickard 1981)</td>
<td>Author use a slightly different form of Hagen-Poiseuille equation with correction factors vary with the shape and with the eccentricity. k=1 (circular cross section); k=1 (mildly circular ellipses cross section); k=1.43 (square cross section); k=1.98 (equilateral cross section)</td>
<td>( g_{ij} = k \frac{D^2_h}{128L_h} )</td>
</tr>
<tr>
<td>(Koplik 1982)</td>
<td>'Ball and Stick' model. Considers contribution of throat entrance and exit effects. Also used in (Jerauld and Salter 1990), Rajaram et al. (1997).</td>
<td>( g_{ij} = \frac{R_t^3}{3 + \frac{8L_t}{\pi R_t}} )</td>
</tr>
<tr>
<td>(Koplik, Lin et al. 1984)</td>
<td>Applies to an elliptical cross section. Also used in (Seeburger and Nur 1984) with additional terms added for deformation due to confining stress.</td>
<td>( g_{ij} = \frac{\pi a^3 b^3}{4L_t (a^2 + b^2)} )</td>
</tr>
<tr>
<td>(Roberts and Schwartz 1985)</td>
<td>Funnel-shaped geometry, using average cross-sectional area</td>
<td>( g_{ij} = \frac{\bar{A}(x)}{L_{RPTP}} )</td>
</tr>
<tr>
<td></td>
<td>( \bar{A}(x) = \frac{1}{2R} \int_{-R}^{R} A(x) )</td>
<td>( \bar{A}(x) = \pi \left[ R_{axis} + R_{ins} - (R_{axis}^2 - z^2)^{\frac{1}{2}} \right] )</td>
</tr>
<tr>
<td>(Dias and Payatakes 1986)</td>
<td>The unit cell is divided into ten compartments to calculate the effective conductance</td>
<td>( g_{ij} = \left[ \sum_{k=1}^{10} \frac{1}{\theta_{frac}} \right]^{-1} )</td>
</tr>
<tr>
<td></td>
<td>( \theta_{frac} = \frac{2\pi (\Delta R_{eff})^2 \sin^2 \theta (1 + 2 \cos \theta)(1 - \cos^2 \theta)^2}{\Delta (R_{eff}^2) \Delta z} )</td>
<td>( T(\theta) = \frac{\cos^2 \theta (1 + 2 \cos \theta)(1 - \cos^2 \theta)^2}{\sin^4 \theta (1 + 3 \cos^2 \theta)} )</td>
</tr>
</tbody>
</table>

Table cont’d
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description/Notes</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constantinides and Payatakes (1989)</td>
<td>Also used in Constantinides and Payatakes (1996).</td>
<td>( g_{ij} = \frac{-\Delta P_i \left( \frac{1}{Z_i' R_{p,i}} \right)}{4\pi R_t^2 R_{p,i}} + \frac{-\Delta P_i' \left( \frac{1}{Z_i' R_{p,i'}} \right)}{4\pi R_t^2 R_{p,i'}} + \frac{8 L_t}{\pi R_t^3} )</td>
</tr>
<tr>
<td>Bryant and Blunt (1992) (Motealleh and Bryant 2007)</td>
<td>Hagen Poiseuille flow using throat effective radius</td>
<td>( g_{ij} = \frac{\pi R_{eff}^4}{8 L_T} )</td>
</tr>
<tr>
<td></td>
<td>Also used in (Bryant, King et al. 1993) with LPTP made smaller to prevent double counting of pore length.</td>
<td>( L_t = L_{PTP} ) ( L_T = L_{RPTP} )</td>
</tr>
<tr>
<td>Goode and Ramakrishnan (1993)</td>
<td>Applies to a four-cusp throat created between four spheres of radius Rs.</td>
<td>( g_{ij} = \frac{8(4 - \pi)^2 A_t R_t^2}{26.42 \pi^2 L_T} )</td>
</tr>
<tr>
<td>Ewing and Gupta (1993)</td>
<td>Based on the formula for creeping flow through a circular orifice.</td>
<td>( g_{ij} = \frac{R_t^3}{3} )</td>
</tr>
</tbody>
</table>
| Bryant, Mellor et al. (1993)                  | Flow path is subdivided into series of conical frusta. The \( \Delta z \) elements run along the axis of the pore throat. The range of the summation in the final expression for \( g_{ij} \) should be shortened somewhat from the LPTP to prevent double counting of throat length as multiple throats enter the same pore. | \( g_{ij} = \left[ \sum_{\Delta z} \frac{1}{\theta_{frac}} \right]^{-1} \)  
\( \theta_{frac} = \frac{2\pi (\Delta R_{eff}) T(\theta)}{\Delta (R_{eff}) \Delta z} \)  
\( T(\theta) = \frac{\cos^2 \theta (1 + 2 \cos \theta) (1 - \cos^2 \theta)^2}{\sin^4 \theta (1 + 3 \cos^2 \theta)} \) |
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description/Notes</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ioannidis and Chatzis 1993)</td>
<td>Applies to a rectangular cross section. Also used by Pan et al., (2001), Talukdar et al. (2002) (with throat resistance only), and (Gostick, Ioannidis et al. 2007).</td>
<td>$g_{ij} = \frac{1}{2} \left( \frac{1}{g_{p,i}} + \frac{1}{g_{p,j}} \right)$</td>
</tr>
<tr>
<td>(Matthews, Moss et al. 1993)</td>
<td>Each pore-throat-pore connection in the network consists of three components: a throat and half of each adjoining pore. The second equation is for gas with slip correction,</td>
<td>$g_{ij} = \frac{57}{32} g_{p,i} + \frac{57}{32} g_{p,j}$ $g_{ij} = \frac{\pi R_t^4}{8L_t} \left[ 1 + \frac{s\lambda}{R_t} \right]$</td>
</tr>
<tr>
<td>(Schlueter 1995, Schlueter, Zimmerman et al. 1997, Lock, Jing et al. 2002, Quinton, Hayashi et al. 2008).</td>
<td>Constricted circular tube. The constriction factor $f_c$ prevents an overestimation resulting from the straight tube assumption. $L$ can be replaced by $2R_s$ term (equal to grain diameter) as a surrogate measure of the throat length. Mean hydraulic radius is computed from a 2D micrograph.</td>
<td>$g_{ij} = \frac{\pi R_t^4}{8L_t} f_c$ $f_c^{\text{sinusoidal}} = \frac{256\gamma^2}{(1 + \gamma)^4(5\gamma^2 + 3\gamma^2 + 3\gamma + 5)}$ $f_c^{\text{sawtooth}} = \frac{48\gamma^3}{(1 + \gamma)^4(1 + \gamma + \gamma^2)}$ $\gamma = \frac{R_{\min}}{R_{\max}}$</td>
</tr>
<tr>
<td>Reference</td>
<td>Description/Notes</td>
<td>Equation</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------</td>
<td>----------</td>
</tr>
<tr>
<td>(Thompson and Fogler 1997)</td>
<td>Term in large brackets is a correction derived from the solution for creeping flow through a hyperbolic Venturi. $R_{CFD}$ computed using FEM. (Fredd and Fogler 1998) used BEM to compute $R_{CFD}$ and (Thompson, Willson et al. 2008) used $Reff$.</td>
<td>$g_{ij} = \frac{\pi R_t^4}{8L_t} \left[ \frac{4L_t(1 + \zeta_0)}{3R_t(\sin \eta_0)^3} \left( \frac{\lambda_R}{\lambda_R + 1 + \tan^{-1} \lambda_R} \right) \right]$ $R_t = R_{CFD}$ or $Reff$</td>
</tr>
<tr>
<td>(Blunt 1997)</td>
<td>Also used by Nilsen et al., (1996), Bakke and Oren (1997), Mogensen and Stenby (1998), Dillard and Blunt (2000), Hughes and Blunt (2000), and Bustos and Toledo (2003).</td>
<td>$g_t = \frac{\pi \left( \frac{1}{2} (R_t + R_{t,ev}) \right)^4}{8L_t}$ $g_p = \frac{\pi \left( \frac{1}{2} (R_p + R_{p,ev}) \right)^4}{8L_p}$ $\frac{1}{g_{ij}} = \frac{1}{g_t} + \frac{1}{2} \left( \frac{1}{g_{p,i}} + \frac{1}{g_{p,j}} \right)$</td>
</tr>
<tr>
<td>(Øren, Bakke et al. 1998) Also used by Patzek and Silin (2001), Patzek (2001), Blunt et al., (2002), Øren and Bakke (2003), Valvatne and Blunt (2004), and DiCarlo (2006).</td>
<td>Mason and Morrow shape factor-based calculations, initially for triangles only. Expanded to other regular shapes (circles, triangles, squares). (Joekar Niasar 2010) developed a technique for continuously recovering shape factor distribution of irregular hyperbolic polygons. Øren later defined effective pore body and throat lengths.</td>
<td>$G = \frac{A}{P^2}$ assumed $A^3 = \frac{r^2}{4G}$ $g = \frac{3A^2 G}{5L}$ $\frac{1}{g_{ij}} = \frac{1}{g_t} + \frac{1}{2} \left( \frac{1}{g_{p,i}} + \frac{1}{g_{p,j}} \right)$ $L_i = L_i^1 \left( 1 - 0.5 \left( \frac{R_t}{R_i} \right) \right)$; $L_i^1$ is the distance between node $i$ and constriction $L_t = L_{ij} - L_i - L_j$ $L_{ij} = \text{length between nodes } i \text{ and } j$</td>
</tr>
</tbody>
</table>

Table cont’d
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description/Notes</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Aker, Jørgen Måløy et al. 1998)</td>
<td>( k ) is the permeability. Also used by Matthews (1993), Knudsen (2002).</td>
<td>( g_{ij} = \frac{\pi R_i^2 k}{L_t} )</td>
</tr>
<tr>
<td>(Sisavath, Jing et al. 2000)</td>
<td>Author’s used approximate solutions from other problems in applied mathematics (torsion of an elastic bar) with the same governing PDE as laminar flow through a pipe of constant cross section. They compared the approximate solutions to CFD solutions for various shapes.</td>
<td>( g_{ij} = \frac{D_H^2 A_t}{32 L_t} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( g_{ij} = \frac{A_t^4}{16 \pi^2 J L_t} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( l = \int [(x - x_0)^2 + (y - y_0)^2] dA )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( g_{ij} = \frac{A^2}{4 \pi L_t \left[ \frac{R_{ins} R_{circ}}{R_{circ} + R_{ins}} \right]} )</td>
</tr>
<tr>
<td>(Liang, Ioannidis et al. 2000)</td>
<td>Authors applied this formula to a voxel image so that ( D_z ) was the voxel size and the summation was applied to each voxel along the pore-throat axis ((-Z ) to (+Z ) in the current notation).</td>
<td>( \frac{1}{g_{ij}} = \sum_{-Z}^{+Z} \left[ \frac{128 \Delta z}{\pi D_H^2 \Delta z} \right] )</td>
</tr>
<tr>
<td>(Blower 2001)</td>
<td>Used for pores formed by bubbles (e.g., in volcanic media), where Rap is the radius of the aperture between two joined bubbles.</td>
<td>( g_{ij} = \frac{\pi R_{ap}^3}{16} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( R_{ap} = \frac{R_{ins}^2 - L_{PTP}^2}{4} )</td>
</tr>
</tbody>
</table>

Table cont’d
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description/Notes</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Sisavath, Jing et al. 2001)</td>
<td>Authors’ derived a closed-form expression for creeping flow through a sinusoidal throat using an asymptotic series solution. The expression in this table is adapted from the original so that it can be written in terms of pore and pore-throat parameters.</td>
<td>( g_{ij} = \frac{\pi (R_p + R_t)^4}{128 L_t} \left( \frac{1}{3} \frac{16 \pi^2 \delta^2 (1 - \delta^2)}{\lambda^2 (2 + 3 \delta^2)} \right) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \lambda = \frac{2 L_t}{R_p + R_t} )</td>
</tr>
<tr>
<td>(Patzek and Silin 2001)</td>
<td>See reference for exact formulas for triangles and ellipses and for restrictions on the admissible triangles.</td>
<td>( g_{ij} = \left( 1 + \frac{1}{\lambda} \right)^2 \left[ 1 - \frac{64}{3} \sum_{n=0}^{\infty} \frac{\tanh (2n + 1) \pi \lambda}{(2n + 1)^{5/2}} \right] \frac{A_i^2}{L_t} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( g_{ij} \approx \frac{1}{2} G \frac{A_i^2}{L_t} ) (ellipse)</td>
</tr>
<tr>
<td>(Thompson 2002)</td>
<td>Designed for high-porosity fibrous materials. Asymptotic expression will diverge for lower porosities.</td>
<td>( g_{ij} = \frac{2}{3} g_t + \frac{1}{3} g )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( g_t = \frac{1}{4 \phi} \left[ - \ln \varphi - 1.476 + 2 \phi - \frac{\phi^2}{2} \right] \frac{R^2_i A_{tot}}{L_t} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( g_\perp = \frac{1}{4 \phi} \left[ - \ln \varphi - 1.476 + 2 \phi - 1.774 \phi^2 + 4.076 \phi^3 \right] \frac{R^2_i A_{tot}}{L_t} )</td>
</tr>
<tr>
<td>(Dadvar and Sahimi 2003)</td>
<td>Author uses hydraulic radii which equal cross-sectional area divided by wetted perimeter.</td>
<td>( g_{ij} = \frac{\pi R_{h,w}^4}{8 L_{PTP}} )</td>
</tr>
<tr>
<td>(Mortensen, Okkels et al. 2005)</td>
<td>Shape-correction to Hagen-Poiseille flow. Hydraulic resistance ((1/g_i)) depends linearly on dimensionless compactness ((1/G)), but for different shapes, the coefficients are nonuniversal. Using ( G ) for consistency.</td>
<td>( g_{ij} = \frac{A^2_i}{\alpha(G)L_t} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \alpha(G) = \frac{8}{3G} - \frac{8 \pi}{3} ) (elliptical shape)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \alpha(G) = \frac{22}{7G} - \frac{8}{3} ) (rectangular shape)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \alpha(G) = \frac{25}{11G} - \frac{40 \sqrt{3}}{17} ) (triangular shape)</td>
</tr>
<tr>
<td>Reference</td>
<td>Description/Notes</td>
<td>Equation</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>(Mortensen, Okkels et al. 2005)</td>
<td>Poiseuille tube flow for harmonically-perturbed circle cross section. Note: $k=2$ yields elliptical shape result</td>
<td>$\alpha(G) = \frac{8}{(1 + k)G} - \frac{8\pi(3 - k)}{(1 + k)} + O(\epsilon^4)$</td>
</tr>
<tr>
<td>(Perrin, Tardy et al. 2006)</td>
<td>Formula is based on Poiseuille flow in a rectangular duct. Applied to modeling flow in an etched micromodel with rectangular channels.</td>
<td>$g_{ij} = \frac{a^3b}{12L_t} \left[ 1 - \frac{192a}{\pi^5b} \sum_{i=1,3,5} \frac{\tanh \frac{j\pi b}{2a}}{j^5} \right]$</td>
</tr>
</tbody>
</table>
| (Li, Peters et al. 2006)         | The conductance is sampled from the log-normal distribution with mean and variance, which depend on the log-volume summation as they assume strong correlation between conductance and adjacent pore volumes | $g_{ij} = N(\mu, \sigma^2)$
$(\mu_X)_{ij} = \mu_X + \rho \frac{\sigma_X}{\sigma_Y} (V_{ij} - \mu_Y)$
$(\sigma^2_X)_{ij} = \sigma_X^2 (1.0 - \rho^2)$
$V_{ij} = \ln V_i + \ln V_j$ |
<p>| (Jia, Dong et al. 2008)          | Jia (2008) gives a formula for arbitrary triangular tubes. The numerically-determined shape factor $s$ is the same for all triangular tubes, 80/3 | $g_{ij}^{arbitrary \Delta} = \frac{4R_{ins}^2 A_t}{s L_t} = \frac{3R_{ins}^2 A_t}{20 L_t}$ |</p>
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description/Notes</th>
<th>Equation</th>
</tr>
</thead>
</table>
| (Ryazanov, Van Dijke et al. 2009) | Flow through n-cornered star-shaped cross sections. $B(n)$ and $C(n)$ are fifth-order polynomials of $n$. E.g. $B(n)=1.4264$; $C(n)=0.0374$ | $g_{ij} = \bar{g} A_i^2$  
$\bar{g} = C(n) \left( \frac{c_{n,y}^2}{G_{n,\text{max}}} \right)^{B(n)}$  
$G_{n,y} = \frac{1}{4} \left( \sin(\frac{y}{n}) n \sin \left( \frac{\pi}{n} \right) \right)^{2}$, $y < \frac{\pi}{2} - \frac{\pi}{n}$ |
| (Sholokhova, Kim et al. 2009)   | $\tilde{g}_i$ and $p$ are best fit parameter values from LBM computed conductance versus shape factor $G$ | $l_{ij} = \frac{w_l t}{g_t} + (1 - w) \left[ \frac{l_t}{g_t} + \frac{l_j}{g_j} \right]$  
$\bar{g} = \tilde{g}_i G^p$ |
| (Sochi 2010)                    | Analytic solutions for Newtonian flow through axisymmetric corrugated tubes (Conical; Parabolic; Hyperbolic; Hyperbolic Cosine; Sinusoidal). $R_{\text{max}}=R_p$, $R_{\text{min}}=R_t$ (Sochi 2013) obtained the same results using 1D Navier-Stokes derived analytical expressions for converging-diverging axisymmetric geometries | $g_{ij}^{\text{Conical}} = \frac{3\pi(R_{\text{max}} - R_{\text{min}})}{8L_t \left( \frac{1}{R_{\text{min}}^2} - \frac{1}{R_{\text{max}}^2} \right)}$  
$g_{ij}^{\text{Parabolic}} = \frac{\pi}{4L_t} \left[ \frac{1}{3R_{\text{min}}^2 R_{\text{max}}^3} + \frac{12R_{\text{min}}^2 R_{\text{max}}^2}{R_{\text{max}}^2} + \frac{5}{R_{\text{max}}^2} + \frac{5 \arctan \left( \frac{R_{\text{max}} - R_{\text{min}}}{\sqrt{R_{\text{min}}^2 - R_{\text{max}}^2}} \right)}{8R_{\text{min}}^2} \right]$  
$g_{ij}^{\text{Sinusoidal}} = \frac{2\pi(R_{\text{max}} R_{\text{min}})^2}{L_t \left[ 2(R_{\text{max}} + R_{\text{min}})^3 + 3(R_{\text{max}} - R_{\text{min}}) (R_{\text{max}} - R_{\text{min}})^2 \right]}$  
$g_{ij}^{\text{Hyperbolic Cosine}} = \frac{3\pi R_{\text{min}}^4}{8L_t} \left[ \frac{\arccosh R_{\text{max}}}{R_{\text{min}}} \right] \tanh \left( \arccosh \frac{R_{\text{max}}}{R_{\text{min}}} \right) \left[ \text{sech}^2 \left( \arccosh \frac{R_{\text{max}}}{R_{\text{min}}} + 2 \right) \right]$  
$g_{ij}^{\text{Hyperbolic}} = \frac{\pi}{4L_t} \left[ \frac{1}{R_{\text{min}}^2 R_{\text{max}}^2} + \frac{\arctan \left( \frac{R_{\text{max}}^2 - R_{\text{min}}^2}{R_{\text{min}}^2} \right)}{R_{\text{min}}^2 \sqrt{R_{\text{max}}^2 - R_{\text{min}}^2}} \right]$ |

Table cont’d
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description/Notes</th>
<th>Equation</th>
</tr>
</thead>
</table>
| (Yang and Balhoff 2017) | Two Conical Frustums, using the analytical velocity solution of Happel and Brenner (2012). In that work on particle deposition, volume of deposited particles was used to update pore and throat radii, assuming all decrease by the same ratio, $x$ | $g_i = \frac{\pi (1 + 2 \cos \theta_0)(1 - \cos \theta_0)^2 L_i^3}{2 \left(1 - \frac{R_i}{R_i}\right)^3 \left[\left(\frac{R_i}{R_i}\right)^3 - 1\right]}$, \quad $i = 1, 2$
$g_{ij} = \frac{g_1 g_2}{g_1 + g_2}$
$x = \left[\frac{V_{throat} - V_{dep}}{V_{throat}}\right]^{\frac{1}{2}}$, \quad $R_i^{\text{New}} = R_i^{\text{Old}} x$, \quad $i = 1, 2, t$ |
### Table 0.2. Variable Descriptions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_s$</td>
<td>Surface area (pore or pore throat)</td>
</tr>
<tr>
<td>$A_t$</td>
<td>Cross-sectional area of the pore throat</td>
</tr>
<tr>
<td>$A_{tot}$</td>
<td>Cross-sectional area of a pore-pore connection (void + solid area).</td>
</tr>
<tr>
<td>$A$</td>
<td>Major semiaxis in the cross section of a non-circular pore throat</td>
</tr>
<tr>
<td>$B$</td>
<td>Minor semiaxis in the cross section of a non-circular pore throat</td>
</tr>
<tr>
<td>$D_h$</td>
<td>Hydraulic Diameter ($D_h = 4A_t/P_t$)</td>
</tr>
<tr>
<td>$f$</td>
<td>Solid volume fraction</td>
</tr>
<tr>
<td>$f_1$</td>
<td>Function used in (Ioannidis and Chatzis 1993), referenced to (Berker, Schaaf et al. 1963)</td>
</tr>
<tr>
<td>$f_2$</td>
<td>Constriction factor (Schlueter 1995)</td>
</tr>
<tr>
<td>$G$</td>
<td>Shape factor (Dimensionless hydraulic radius): $G=A_t/P_t^2$, unless otherwise noted</td>
</tr>
<tr>
<td>$g_{ij}$</td>
<td>Overall hydraulic conductance of the connection between pores $i$ and $j$.</td>
</tr>
<tr>
<td>$g_p$</td>
<td>Hydraulic conductance of the pore section of a subdivided connection</td>
</tr>
<tr>
<td>$g_t$</td>
<td>Hydraulic conductance of the throat section of a subdivided connection</td>
</tr>
<tr>
<td>$k$</td>
<td>Order of the harmonic perturbation. Integer &gt;2</td>
</tr>
<tr>
<td>$L_t$</td>
<td>Throat length (generic)</td>
</tr>
<tr>
<td>$L_{PTP}$</td>
<td>Throat length (pore to pore distance)</td>
</tr>
<tr>
<td>$L_{RPTP}$</td>
<td>Revised throat length</td>
</tr>
<tr>
<td>$l_1$, $l_2$</td>
<td>Near and far distances from the ends of the tube to the apex of the cone</td>
</tr>
<tr>
<td>$l_p$</td>
<td>Pore length for a subdivided connection</td>
</tr>
<tr>
<td>$l_t$</td>
<td>Throat length for a subdivided connection</td>
</tr>
<tr>
<td>$P_t$</td>
<td>Perimeter of the pore throat</td>
</tr>
<tr>
<td>$P_w$</td>
<td>Wetted perimeter of the pore throat</td>
</tr>
<tr>
<td>$R_{ave}$</td>
<td>Average radius of bounding spheres</td>
</tr>
<tr>
<td>$R_{axis}$</td>
<td>Radius (average) of the bounding spheres along the axis of a throat</td>
</tr>
<tr>
<td>$R_{circ}$</td>
<td>Circumscribed radius</td>
</tr>
<tr>
<td>$R_{CFD}$</td>
<td>Throat radius determined from CFD solution of laminar flow through a tube of the throat's cross section.</td>
</tr>
<tr>
<td>$R_{eff}$</td>
<td>Effective radius: $Reff = \frac{1}{2}(R_{ins} + Req)$</td>
</tr>
<tr>
<td>$Req$</td>
<td>Equivalent-area (or equivalent-volume) radius</td>
</tr>
<tr>
<td>$R_f$</td>
<td>Fiber radius</td>
</tr>
<tr>
<td>$R_{h,w}$</td>
<td>Hydraulic radius (wetted): $Rh,w = 2At/P_w$</td>
</tr>
<tr>
<td>$R_{ins}$</td>
<td>Inscribed radius</td>
</tr>
<tr>
<td>$R_p$</td>
<td>Pore radius (generic)</td>
</tr>
<tr>
<td>$R_t$</td>
<td>Throat radius (generic)</td>
</tr>
<tr>
<td>$V$</td>
<td>Volume (pore or pore throat)</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Angle of taper</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Cross-sectional aspect ratio of a non-circular pore throat ($l = a/b$)</td>
</tr>
<tr>
<td>$\lambda_R$</td>
<td>$R_{axis}/cfcv$, where cfcv is the distance to the focal circle of a hyperbolic venturi.</td>
</tr>
</tbody>
</table>
VITA

Timothy Wayne Thibodeaux was born, raised, and educated in Louisiana. He is from the town of Lockport, Louisiana in Lafourche Parish. After graduating from Central Lafourche High School in 2006, he spent the next four years earning a B.S. in Chemical Engineering at the University of Louisiana at Lafayette, graduating in 2010. He began pursuit of a doctorate in Chemical Engineering at LSU thereafter. In 2013, he was married to Madison Thibodeaux (née Lowrey). They live in New Orleans, LA with their young daughter, Olivia. Timothy is a devout Roman Catholic. He enjoys traveling and spending time with his wife, playing with his daughter, numerous outdoor activities, and discovering and playing music.