Estimation of relative transport properties of fuel cell and electrolyzer transport layers by pore network and continuum based direct simulations

by

Seongyeop Jung

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

University of Alberta

Abstract

Improvements in the imaging techniques have enabled the capturing of images with high resolution, thereby permitting the reconstruction of complex porous media that can be analyzed by computer simulations. The two most popular methods for numerically analyzing transport in porous media are pore network modeling (PNM) and direct numerical simulation (DNS).

This thesis describes a method to characterize the microstructure of various porous media used for fuel cell and electrolyzer. In this work, two fibrous gas diffusion layers (Toray 120C, SGL 39BA) and one powder based sintered titanium paper were directly scanned using X-ray micro computed tomography (μCT). The obtained raw images were binarized, and cropped to three subsamples. To assess the validity of subsamples, the statistical characterization were performed using two-point correlation and chord length functions. Stochastic reconstructions were used to generate catalyst layers (CLs).

The microstructure of the subsamples are analyzed by computing pore size distribution (PSD), mercury intrusion porosimetry (MIP), and water intrusion. PNM uses pore networks extracted from the subsamples to obtain PSDs. The networks are then used for non-wetting phase intrusion in the domain using percolation algorithms. For DNS, the sphere fitting algorithm is used for the PSD calculations. The PSD information is then used for water and mercury intrusion using the cluster based full morphology (FM) algorithm. PSDs computed from MIP data are in good agreement with the PSDs from sphere fitting algorithm.

Dry and wet transport properties are computed using both methods and compared to experimental references. Two diameter cases provided by PNM are used for numerical calculations. Using the equivalent diameter returned closer results compared to the experimental transport properties. In PNM, relative water permeability results show different trends based on the intrusion algorithm used for the transport simulations. Employing the invasion percolation (IP) algorithm returns better results compared to the reference results. The DNS results obtained based on saturated images return acceptable results.

Keywords: Pore network, Full morphology, Direct numerical simulation, Image analysis, Tomography, Two-phase flow, Mercury intrusion porosimetry, Pore size distribution

Acknowledgements

I would like to thank Dr. Marc Secanell for providing me the opportunity to pursue my research under his support. His academic guidance, dedication, and critical thinking have been truly inspirational. It has been a pleasure to meet a supervisor who has shown a true enthusiasm for his work. Not only have I learned about my research, but also attitude toward life. My special thanks to Professor Jeff Gostick for providing additional support and guidance.

I would like to thank my colleagues in the Energy Systems Design Lab, for their help and making an amazing experience. You made my life in Edmonton very easy and fun. Special thanks to Fei for helping me in experimental supports, and Alex for his dedicated efforts for setting up OpenFCST simulations for me.

Finally, I'd like to thank my family who have given me support during my degree. In particular, I like to thank my wife for her encouragement and love throughout.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Today's energy transition is one related to a move away from the consumption of carbon-emitting fuels to the consumption of non-carbon-emitting fuels, a process called decarbonization [4]. While the current energy transition is being driven by technological, economic, and political factors, it is primarily driven by environmental factors based on changing social values - the drive to reduce global greenhouse gas (GHG) emissions. [4].

Canada generates 1.7 % of global GHG emissions, however it is one of the most energy and emission intensive nations in the world. The International Energy Agency (IEA) recommends that Canada take action to reduce its emissions and energy intensities in order to strengthen its position as a responsible energy supplier and user [4]. The Organization for Economic Co-operation and Development (OECD) notes Canada's status as one of the most GHG emission-intensive economies in the world, and the fourth largest emitter of GHGs in the group of OECD nations [4]. Transportation represents 25% of overall GHG emissions. Growing economic activity in Canada affects the number of freight trucks on the road, thus emissions from the freight transportation subsector are projected to rise [5].

Polymer electrolyte membrane (PEM) fuel cells are electrochemical energy conversion devices which have proven to be one of the most promising technologies owing to their high efficiency, zero local emissions, and rapid start-up capacity. In addition to transportation, fuel cells are being applied in many applications such as a backup power, forklift engines, and power plants [6].

Figure 1.1 shows the schematic of a proton exchange membrane fuel cell (PEMFC). A PEMFC consists of four main components: a) the bipolar plate, which contains the current collectors and gas channels; b) gas diffusion layer (GDL) and micro porous

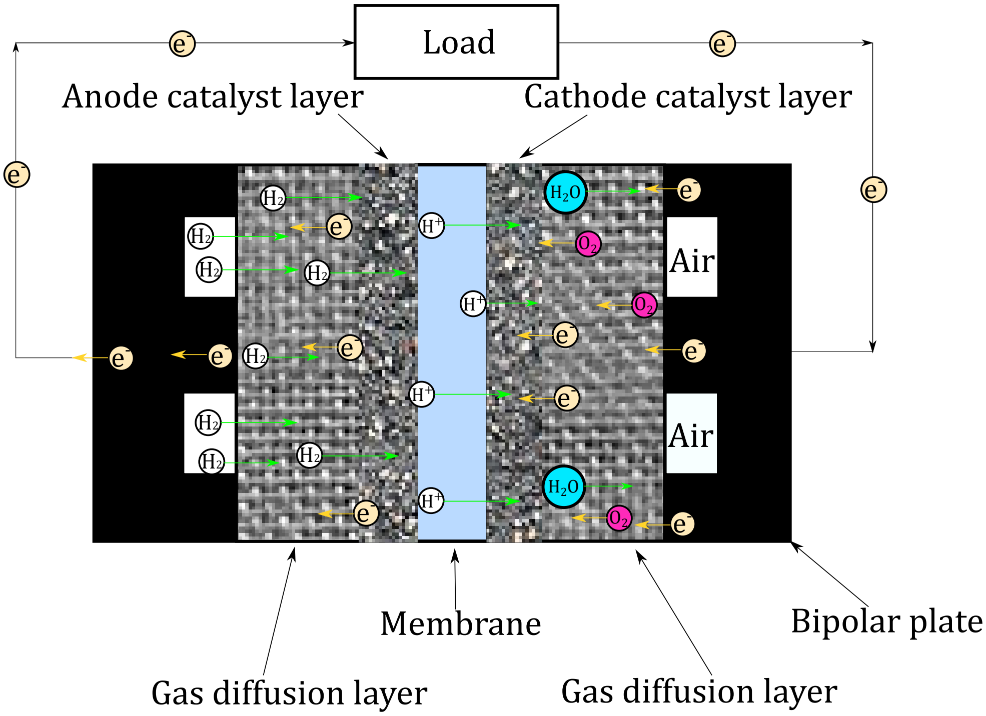


Figure 1.1 – Schematic of a proton exchange membrane fuel cell.

layer (MPL); c) the catalyst layer (CL); and d) the proton exchange membrane. In a PEMFC, hydrogen enters through the anode channels and reacts to produce protons and electrons. Protons are conducted through the polymer electrolyte membrane (PEM) to the cathode side. The protons react with the oxygen and electrons from the cathode side to produce water and heat. The electrochemical reactions for cathode and anode can be shown as:

$$\begin{aligned} \text{Anode}: \; & H_2 \rightarrow 2H^+ + 2e^- \\ \text{Cathode}: \; & \frac{1}{2}O_2 + 2H^+ + 2e^- \rightarrow H_2O \end{aligned} \tag{1.1}$$

where the liquid water is the final product of the reaction. The overall electrochemical reaction in the PEMFC can be written as:

$$\frac{1}{2}O_2 + H_2 \rightarrow 2H_2O + Electricity + Heat \tag{1.2}$$

Effective water management mitigates mass transport losses and leads to an increase in the maximum current density and power output. At high current density, water vapour pressure may exceed saturation pressure within the cell, leading to the formation of liquid water. This water blocks gas transport and reduces the maximum power output of the cell. A highly porous and hydrophobic GDL in the electrode aids the supply and removal of reactants and products, but the water accumulation

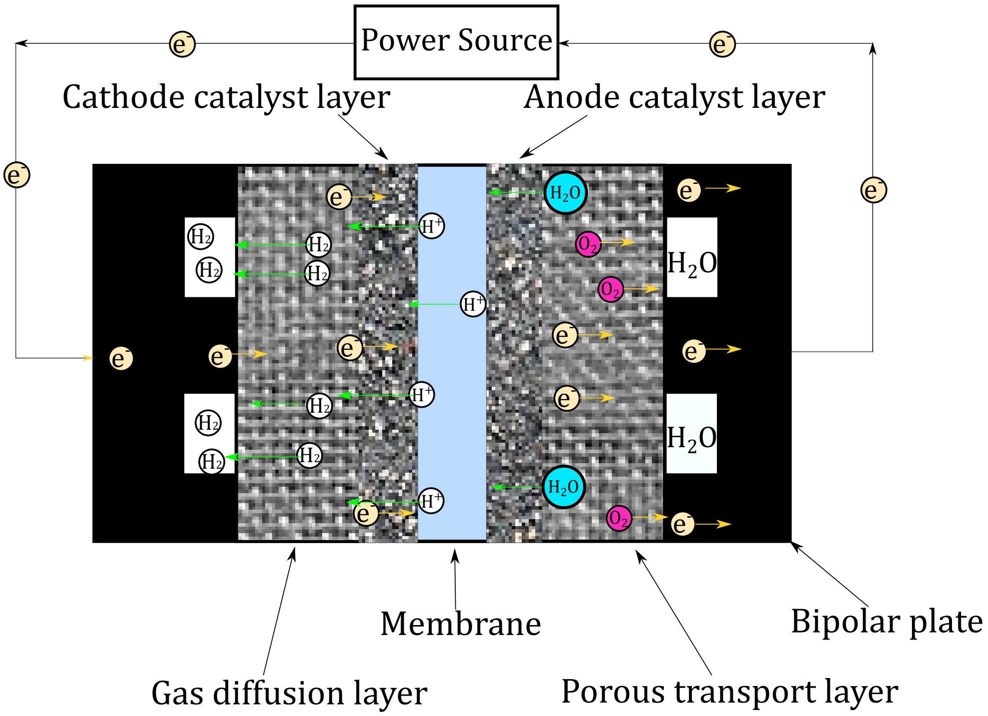


Figure 1.2 – Schematic of a polymer electrolyte water electrolyzer.

within the GDL pore region leads to a decrease of pore sizes. The pores then need higher pressure to be intruded by gas. Therefore, understanding water balance of high current density in the cell is crucial to cell performance.

PEMFC are powered by hydrogen gas. However, hydrogen is not found in nature, and it must be manufactured. A technology to manufacture hydrogen from water is the polymer electrolyte water electrolyzer (PEWE). Figure 1.2 shows a schematic of PEWE. The half reactions at the anode and cathode can be shown as:

$$\begin{aligned} \text{Anode}: \ & H_2O \rightarrow \frac{1}{2}O_2 + 2H^+ + 2e^- \\ \text{Cathode}: \ & 2H^+ + 2e^- \rightarrow H_2 \end{aligned} \tag{1.3}$$

In the PEWE, water is supplied to the anode side and is transported through porous transport layer (PTL) to the catalyst layer. It is decomposed into protons, electrons and oxygen. The oxygen is then transported through the anode PTL to be discharged from the cell. Protons are transported through the proton exchange membrane (PEM) to the cathode, while electrons are transported through the external circuit. The anode half reaction is called the oxygen evolution reaction (OER) and the cathode half reaction is called the hydrogen evolution reaction (HER). Mass transport losses are due to the accumulation of oxygen gas in the anode, which might displace the reacting liquid water from the catalyst surface. The oxygen gas generated from the anode catalyst layer (CL) should be removed from the anode electrode but it accumulates in the anode CL and PTL. The oxygen bubbles trapped in the anode PTL hinder the mass transport efficiency so that the mobility of the reactant is limited.

Given the critical importance of mass transport in enhancing fuel cell and PEWE performance, the effect of the morphology on gas and water transport in these materials need to be studied in detail.

1.2 Background

1.2.1 Porous media

Proton exchange membrane fuel cells (PEMFCs) and polymer electrolyte water electrolyzers (PEWEs) are composed of an anode, a cathode, and a membrane. The anode and cathode are the heart of the cell and both are porous media. The major components of anode and cathode are the catalyst layers (CLs), and the gas diffusion layers (GDLs) in PEMFC; and the catalyst layers (CLs) and the porous transport layers (PTLs) in PEWE. A catalyst layer is the centre of the electrochemical reactions. Reactants and products are delivered through the layers and electrochemical reactions take place in CL. The anode and cathode CLs are responsible for the hydrogen oxidation reaction (HOR) and oxygen reduction reaction (ORR) in PEMFC; and, the oxygen evolution reaction (OER) and the hydrogen evolution reaction (HER) in PEWE. The anode and cathode are separated by PEM in both PEMFC and PEWE. Each of the components has a specific function and material characteristics, which are discussed in the next sections.

1.2.1.1 Catalyst layers

Catalyst layers (CLs) in PEMFC and PEWE, where the electrochemical reaction occurs, are porous composites made of ionomer, and either carbon supported catalyst or iridium particles, respectively. It is a thin film with nanoscale pores. The pores enable the reactants and products transport to the reaction sites. The typical material properties for CL are shown in Table 1.1.

Table 1.1 – Parameters of CL

Thickness	5 – 60 μm [7]
Pore size	5 – 130 nm [8]
Porosity	30 – 60 % [9]
Pt particle size	2 – 5 nm [8]

New fabrication techniques, such as electrospraying and inkjet printing, have enabled researchers to develop structures with varying porosity and pore sizes. The question, however, remains as to what is the most appropriate structure to be used in each application and how the morphology affects transport and water accumulation. In order to address this question, the development of numerical models for computer-aided design of porous media is of paramount importance.

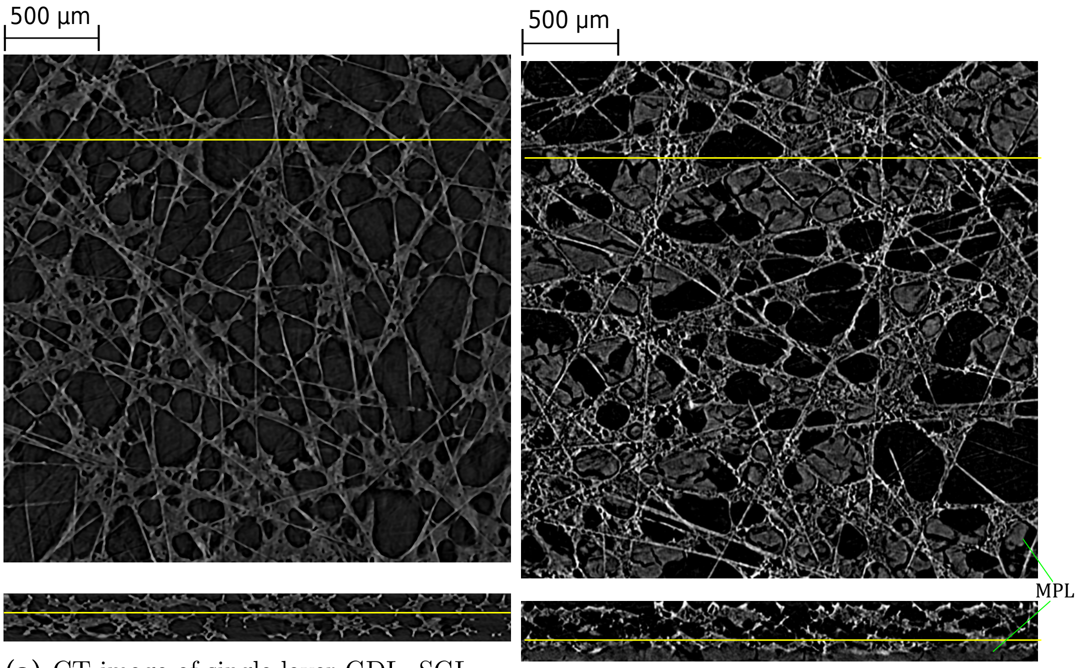
1.2.1.2 Fuel cell gas diffusion layers

Gas diffusion layers (GDLs) in PEMFC and porous transport layers (PTLs) in the cathode of PEWE are in direct contact with the gas channels. They are responsible for transporting the fuel and reactant from the channels in the bipolar plates to the catalyst layer, transporting electrons from the CL to the bipolar plates, and providing structural support to the membrane electrode assembly (MEA). They are made of carbon fibers or carbon cloth materials, and usually have high porosity for easy transport of gases. In PEMFCs, they are often treated with PTFE to make them hydrophobic which facilitates water removal from the cell. Nowadays, micro porous layers (MPLs) are often embedded onto the GDL surface, since it has been found that the addition of an MPL at the GDL-CL interface improves the performance of the cell under wet conditions. Carbon based GDLs are generally classified as either single layer or dual layer on the basis of the existence of a micro porous layer (MPL). Figure 1.3 shows comparison of the μCT images between the single and dual layer GDLs.

Conventional GDLs such as SGL 38 and 39 series are manufactured by the following steps [10]:

1. Chopped carbon fibers are processed to a primary carbon fiber web.
2. The raw paper is impregnated with carbonizable resins.
3. They are cured and graphitized to adjust the porosity and to enhance electrical and thermal conductivity.
4. The carbon paper is treated with PTFE to make the material hydrophobic.
5. Micro-porous layer is coated on the carbon paper
6. Sintering bonds the substrate and MPL.

The carbon fiber GDLs typically have a porosity of 70% to 90% with a mean pore diameter on the order of 10 μm [11], and thickness ranges from 100 μm to 300 μm [12].



(a) CT image of single-layer GDL, SGL 38BA

(b) CT image of dual-layer GDL, SGL 38BC

Figure 1.3 – CT images of single and dual layer GDLs in two directions. Carbon based GDLs are classified as either single layer or dual layer on the basis of the existence of a micro porous layer (MPL)

1.2.1.3 Electrolyzer porous transport layers

At the anode side of a PEWE, a titanium (Ti) porous media is a preferred material because of the highly corrosive environment in the anode. Carbon materials used as gas diffusion layers in fuel cell cannot be used in PEWE. The high electrical potentials imposed at the anode side ($\sim$ 2 V) with respect to H_2 electrode, and highly acidic ($\sim$ pH 2) and oxygen rich environment facilitate the oxidation of carbon or graphite materials. Ti is able to withstand the harsh environment and provides effective electrical conductivity. There are three types of Ti PTLs, powder based sintered Ti, felt type Ti fiber paper, and Ti foam [13]. Sintered powder PTLs generally have porosities below 50% whereas the felt type Ti papers are available in a wide range of fiber sizes and porosities [13].

The powder based Ti PTLs, typically fabricated by thermal sintering, have the following properties shown in Table 1.2.

Table 1.2 – Parameters of powder based sintered Ti PTLs for PEWE [1].

Thickness	0.8 – 2 mm
Porosity	20 – 50 %
Pore size	5 – 30 μm
Particle size	25 – 250 μm

1.2.2 Transport properties

Mass transport in PEMFC and PEWE has an important role in the cell performance. The reactants should be transported sufficiently fast to the reaction sites for ideal cell performance as insufficient supply of reactants limits the maximum achievable current density. In PEMFC, GDLs allow transport reactants and products to the catalyst layer. Transport of reactants in the through-plane direction is mainly controlled by diffusion, whereas in-plane transport is both diffusion and pressure driven [14]. In PEWE, transport occurs mainly due to pressure gradients. Accurate determination of these transport properties, i.e., diffusivity and permeability, in both in-plane and through-plane directions under dry and partially saturated conditions is therefore crucial to optimize cell performance in computational modeling.

1.2.2.1 Effective diffusivity

Effective diffusivity is an important parameter for porous layers as it is the main mode of gas transport in porous media of PEMFC. The diffusion of gases to the catalyst layer in the cathode electrode should be as high as possible to achieve high current density in a fuel cell. Diffusion is governed by Fick's law:

$$\mathbf{J} = -D\nabla C \tag{1.4}$$

where $\mathbf{J}$ is the diffusion flux, C is the concentration, and D is the diffusion coefficient.

In porous media, Fick's law must be volume-averaged leading to the average form of equation (1.4):

$$\hat{\mathbf{J}} = -D^{eff}\nabla C \tag{1.5}$$

where $\hat{\mathbf{J}}$ is the superficial flow rate. The effective diffusion coefficient, D^{eff}, can be defined as a function of the bulk diffusion coefficient of gases, porosity, and liquid water saturation of porous media:

$$D_{eff} = D_{bluk}a(\varepsilon)b(S) \tag{1.6}$$

where a(ε) and b(S) are functions defining the impact of the porosity and liquid water saturation on the diffusivity, ε is the porosity, D_{bulk} is the bulk diffusivity. There are some analytical formulas to relate the effective diffusivity to the other properties, however their accuracy is limited, for example, the Bruggman equation is most widely used formula [15]:

$$D_{eff} = D_{bulk}(\varepsilon)^{1.5}(1-S)^{1.5} \tag{1.7}$$

This relation however was developed for a packed bed of spherical particles and therefore it is not applicable to fibrous media or porous media in general.

1.2.2.2 Permeability

Permeability is an important property in electrolyzer and fuel cell operation as it is a material-specific characteristic which is measuring the resistance to flow. For flow at low Reynolds number (Re < 1), flow in porous media can be described by Darcy's law:

$$\nabla p = -\frac{\mu}{k}\mathbf{v} \tag{1.8}$$

where ∇p is the pressure gradient across porous media, $\mathbf{v}$ is the velocity, μ is the dynamic viscosity, and k is the permeability of the porous media. In this flow regime, viscous interactions of the fluid lead to the pressure drop through the porous media.

1.3 Literature review

Many experimental and numerical studies revealed the importance of understanding the microstructure and wettability of the porous media in order to estimate effective transport properties and electrochemical performance [16–23]. For example, varying PTFE content, leading to different pore size distribution (PSD), has been shown experimentally to impact PEMFC performance [23, 24]. However, conducting experiments is time consuming and expensive. Moreover, it is difficult to visualize the transport phenomena inside the cell.

Numerical models can be used to perform careful parametric studies to elucidate the importance of pore-level features, such as PSD, fibre diameter, and many other design parameters. Numerical approaches to estimate transport properties in PEMFC and PEWE materials are: a) pore network models (PNM) [3, 25–28], and b) direct numerical simulations (DNS) [29–33].

This chapter provides a concise review of literature relevant to the works done in this thesis. Specifically, imaging techniques and numerical approaches used for water intrusion and transport properties are highlighted.

1.3.1 Numerical characterization

A popular approach to analyze the aspect of GDL and PTL morphology and transport is pore scale modelling. Pore scale models resolve the heterogeneity of the porous media by solving related governing equations at a pore size scale. Two major pore scale modelling approaches are pore network modelling (PNM) and direct numerical simulation (DNS) [34, 35]. PNM utilizes a simplified network comprised of spherical pores and cylindrical throats. One can either generate an artificial pore network or extract the network from microstructural images, such as μCT tomography. Transport inside the network is computed solving 1D analytical solutions of the relevant transport equations [36]. DNS, on the other hand, employs the porous media geometry obtained from tomography images directly for its simulations. Instead of extracting idealized networks, DNS solves the transport equations on computational meshes generated from tomography images. In voxel-based mesh generation, the image voxels are directly converted to mesh elements. Additional structural information can be exploited for each simulation, but it needs more computational resources than PNM. Each DNS simulation needs a computational mesh to analyze transport properties in porous media. Therefore, analyzing the transport properties of partially-saturated materials requires a new mesh for each saturation level.

1.3.1.1 Pore Network Modeling

Many authors have employed pore network modeling (PNM) to estimate porous media properties, such as diffusivity, permeability, capillary pressure curves, and interfacial area. The basic concept of PNM is to simplify the void space by an idealized geometry of pores and throats, and compute the transport of fluid between pores and throats using simplified governing equations. Because of the simplicity and the computational efficiency, PNM has been widely used for analyzing PEMFC and PEWE porous media.

The first step of the PNM simulation is defining a network that can represent the material of interest. Obtaining a network can be done either generating an artificial network or extracting one from a binary image stack. One approach of modelling PTLs with PNM is using a regular cubic network, i.e. placing pores and throats in a cubic lattice [25, 37]. Achieving high porosity ($\geqslant 80\%$) is, however, difficult when using spherical pores and cylindrical throats in a cubic network because the pores are likely to overlap with one another. For this reason, Gostick et al. [25] computed transport properties of Toray and SGL GDLs using a cubic network with cubic pores and cuboid throats. The size of the rectangular pores and throats were set using a Weibull distribution. The size distribution was calibrated to achieve the desired porosity and the dry transport properties of the commercial PTLs. Material-specific relative gas and liquid permeabilities and diffusivities were computed as a function of water saturation. Water intrusion was simulated by injecting water in the through-plane direction based on the capillary pressure of each throat. They generated cubic networks for two different fibrous GDLs, i.e., SGL 10BA and Toray 090, and showed very close simulation results for mercury intrusion cumulative curves, and dry and wet transport properties compared to experimental data. Cubic networks however are not the most appropriate network to represent the entangled fibrous structure of PTLs.

To overcome this limitation, some research focused on generating more realistic networks. Sinha et al. [38] built a random tetragonal pore network based on the randomly stacked regular fiber screens reported in [39]. Using cubic and cuboid pores and throats, respectively, the pore and throat sizes were assumed to have a cut-off log-normal distribution. The randomly stacked fiber screens and the tetragonal connectivity enabled the pores to be randomly distributed in the 3D domain. The pore network was generated to have similar mean pore diameter, thickness, and porosity to Toray carbon paper without PTFE content. They computed the dry gas permeabilities of the pore network and showed good agreement with experimental data. They also showed the liquid water transport in the pore network; but it was not compared to the experimental data, and the transport properties were not computed.

Luo et al. [40] proposed a topologically equivalent pore network (TEPN). Instead of making a random pore network, they employed a stochastic modelling method to generate pore scale 3D images of carbon paper and carbon cloth GDLs. The generated images were used to extract topologically equivalent pore networks using the maximal ball algorithm developed by Dong and Blunt [41]. The network extraction from the stochastic images allowed generating a pore network with a high porosity, a large span of pore size distribution, and more realistic connectivity between pores compared to a random cubic network. The pore networks were generated with structural inputs from Toray 060 and E-TEK carbon cloth materials. They computed permeabilities under the dry and wet conditions, and the dry permeabilities from the two networks showed good agreement with experimental data. The relative permeability curves were, however, not compared to experimental data or any other literature.

Subsequently, Gostick [42] developed a 3D irregular pore network model using Voronoi and Delaunay tessellations to achieve topological equivalence to Toray 090. Base points were distributed in the model domain, and Delaunay cells were created by selecting three points. Centroids of each Delaunay cell were connected creating the Voronoi tessellation. The Delaunay tessellation represents the pore space and the

Voronoi tessellation represents the solid fiber structure. To account for the anisotropy of fibrous GDL materials, the base points were scaled prior to performing the tessellations. Tranter et al. [26] studied the effect of compression using artificial Voronoi and Delaunay networks. The network was generated to achieve microstructural equivalence to Toray 120. Their simulation results for the dry transport properties showed good agreement to experimental single phase transport data. The effect of compression was to reduce effective transport, especially in the in-plane direction. The model proved it possible to deform the networks to analyze the effect of compression using PNM. For the relative transport properties, both research showed the air relative diffusivity and permeability decay exponentially according to the power relation, $(1-s)^{\lambda}$, with $2 \leq \lambda \leq 4$. The water relative permeability found to fit a log normal expression similar to that presented by Hwang and Weber [2]. The relative transport properties were, however, not compared to another simulations or experimental data.

Another more realistic way of generating pore networks is extracting them from images obtained from μCT images. Gostick [43] developed a network extraction algorithm from 3D microstructure images. The pore regions of the image were first segmented using the watershed segmentation algorithm and the segmented regions were then used to separate the pore network. Pore sizes were determined from the maximum sphere sizes that can fit in pore spaces. Connectivity between pores wase defined by the pore spaces having shared facets. Throat sizes were determined from the maximum circle sizes fit in the facets. They compared the simulation results of Berea sandstone to the network extracted from the maximal ball algorithm by Dong and Blunt [41] and their permeability results showed great agreement with each other. Furthermore, they extracted a network from tomography images for Toray 120A and computed the dry permeability. The permeability values were within acceptable range to their experimental data.

Image-based pore network extraction enabled the use of pore network modeling for a variety of different materials. Since then many authors have employed the network extraction method in PEMFC and PEWE studies. Fazeli et al. [27], using an extracted network from synchrotron based X-ray images, investigated the effect of compression on liquid water transport and oxygen transport resistance, under dry and wet conditions of Toray 090 and SGL 25BC at each compression state. Invasion percolation simulations were performed for the water saturation on the corresponding network of each sample. They found that Toray 090 under 10% compression showed the lowest average saturation and wet transport resistance, whereas saturation was the least at its uncompressed state in SGL 25BC. Lee et al. [44] developed a model for stochastic reconstruction for sintered titanium powder PTL based on the information from μCT images. They used a material density map of the binary images to determine the position of spherical particles. The particles were used as seeding points, and simulated the surface morphology to achieve the irregular morphology of the sintered Ti by filling in the regions surrounding the Ti particles. Pore networks were extracted from the stochastic Ti models with three different mean pore diameter cases and PNM was used to examine the impact of the PTL microstructure on the transport properties [28]. They tested varying inlet conditions for oxygen gas percolation through displaced liquid water in the domain, and found that a small coverage of the inlet face led to high gas saturations, which affect the cell performance. They also reported larger pore and throat diameters exhibits enhanced water permeation.

Despite the amount of research on GDL analysis using PNM extracted from μCT images, a critical gap in literature is the lack of articles that have studied the accuracy of the method by direct comparison of the predicted results to experimental data and direct numerical simulations of μCT images. In this work, the network extraction algorithm developed by Gostick [43] was used to simulate non-wetting phase intrusion and to compute transport properties under the dry and wet conditions.

1.3.1.2 Direct Numerical Simulation

Analysis of pore-scale phenomena has gained popularity over the past decade. The idea solving the species transport equations directly on a computational mesh generated from material images enabled to study pore-scale transport and reaction phenomena. This approach is sometimes called a direct numerical simulation (DNS). Advancements in imaging techniques have enabled visualizing the microstructure of porous layers with high resolution. Numerical tools have been developed to convert μCT, nano-CT, and FIBSEM images into a computational mesh and advances in high performance computing allow now users to perform numerical simulations in domains with many cells enabling the use of a continuum approach. The mesh from images can be generated either directly converting the image voxels into hexahedral elements or using a meshing algorithm to generate tetrahedral elements. The former approach is used both in CFD, and in lattice Boltzmann methods (LBM). The latter approach is somtimes used for CFD simulations.

Wang et al. [45] developed a DNS model to achieve a pore-level description of a PEMFC electrode. They constructed idealized two and three dimensional regular microstructures to represent the cathode catalyst layer. The governing partial differential equations were discretized using the commercial CFD software Fluent. This was the very first implementation of DNS on idealized simple 2D and 3D CL microstructures. Mukherjee et al. [46] applied the same model to more realistic cases. They studied a pore-scale description of species and charge transport through a bilayer cathode catalyst layer (CL). The CL structures were generated using a stochastic reconstruction technique. To make the CL structures statistically homogeneous and isotropic, the porosity and the two-point auto-correlation functions of the image where the same as that of a two-dimensional TEM image of an actual CL sample.

Sabharwal et al. [29] employed FIBSEM image reconstructions and converted them into computational meshes to predict transport properties and electrochemical performance of low loading fuel cell electrodes. The FIBSEM data was converted to a mesh and the gas transport in the CL was studied using Fick's law, i.e., equation (2.13). They showed that the gas transport results were affected by the artificial anisotropy in the microscope images due to FIB slicing. A study of the representative elementary volume showed that, in CLs, gas transport was not affected up to domain sizes of less than 200 nm.

Pfrang et al. [47] computed the thermal conductivities of the three commercial GDLs using μCT images. To compute the thermal conductivity, they employed a commercial software GeoDict. Their results showed clear anisotropy between the in-plane and through-plane directions when it comes to a fibrous carbon paper. They did not distinguish PTFE from carbon fibres because the simple thresholding method used to extract the solid phase could not separate them.

Rosén et al. [48] reported saturation dependent gas transport properties in GDL using μCT images. They computed gas transport properties, i.e. permeability and diffusivity, using lattice Boltzmann and finite difference methods. A synchrotron imaging was used to obtain real-time water saturation images. They found that the saturation in the GDL domain was not homogeneous during the cell operation and water was likely to reside in the larger pores. They computed relative gas permeability and diffusivity in the saturation range of 0.1–0.4. The results were shown to follow power law relationships in the form of $(1-s)^\lambda$ with λ of 3 for the permeability, and λ of 2 for the diffusivity.

Garcia-Salaberri et al. [31, 49] studied effective transport properties of dry and wet cases. The GDL images were obtained from a synchrotron imaging technique. They also tried to estimate the minimum domain size, i.e., representative elementary volume, of fibrous GDLs, that can provide repeatable results. The governing equations were solved by means of the lattice Boltzmann method. To obtain the saturated GDL images, water was invaded into the GDL sample from the bottom face through a PTFE pipe, while air was allowed to be discharged through the top face. In-plane and through-plane relative diffusivity showed following a power law of $(1-s)^\lambda$ with

λ equal to 2 in the IP case, and in the range of 2 to 4 in the TP case in agreement with Rosén et al. [48]. Moosavi et al. [33] employed the same water injecting and image scanning methods, used by Garcia-Salaberri et al. [31], to obtain the saturated GDL images. Instead of using LBM, they utilized a finite volume method (FVM) with tetrahedral mesh elements to investigate the transport properties based on both dry and wet μCT of fibrous GDLs. Their relative transport properties were in good agreement with previous numerical [31, 48] and experimental results [2].

The above researches show that using direct imaging methods to obtain real images of porous media provides sufficient quality to characterize the microstructure of porous media. Computing transport properties with a mesh generated from the real structure images also provides reliable results for both dry and wet conditions.

In order to study transport in partially-wetted samples, understanding the water distribution in the sample is critical. Several research studies performed water intrusion experiments in a synchrotron and imaged the water distribution at varying saturation levels in the porous media. This technique provides an excellent option to study liquid water transport. However, the accessibility of the device is limited and scanning various saturation levels using direct imaging is time consuming. To circumvent this endeavour, the full morphology algorithm has been recently used to predict the water distribution in partially saturated PTLs [30, 50–54].

The full morphology (FM) method, also known as a morphological image opening (MIO), takes a dry binary image of a porous media and uses a structural element to apply the image opening technique. A distinct advantage of FM over PNM is the direct use of tomography images without geometric abstraction. The image opening fills up a sections of the image denoted as pore region with a structural element, usually a sphere. The pore space occupied by the structural elements represents the region saturated by water and the region unoccupied by the elements represents the fraction of dry phase which is air. The water saturated sections can be analyzed by grouping them as a bunch of connected clusters.

Schulz et al. [55] employed FM approach for modelling two-phase behaviour in the gas diffusion medium. They applied morphological opening to determine where the structural element can fit within the pore space. The radius of the structural element was changed based on corresponding capillary pressure. Their model predicted a capillary pressure-saturation curve in good agreement with experimental data; however, their model would not account for the actual GDLs with multiple contact angles.

To compare the intrusion simulations between PNM and FM, Agaesse et al. [53] performed two-phase simulations using PNM and FM in a 3D GDL microstructure obtained from μCT reconstruction. They compared the simulated water saturation results to the experimental μCT reconstructions of partially saturated GDLs at different capillary pressures. They found that both approaches predicted well the experimental liquid water intrusion curves. They computed the effective diffusivity and the effective electrical conductivity using DNS and PNM [56]. They studied two different types of GDL and two compression levels. They compared the effective transport properties obtained from DNS, PNM extracted networks, and PNM structured network (cubic) to experimental data. The DNS results showed the best accuracy, and the structured network cases showed the largest deviations. However, they did not perform relative transport simulations.

Gostick [42] applied both the morphological image opening (MIO) and PNM percolation algorithms to the estimation of capillary pressure-saturation curves. Instead of using dilation and erosion, distance transforms were used for a cluster labelling of invaded regions. They outlined two limitations of the MIO algorithm: a) coalescence of menisci is not accounted for, and b) the intruding water front is limited to spherical shape, thereby implicitly assuming a contact angle of 180°. They performed mercury and the liquid water intrusions on the artificially generated images, and the intrusion cumulative curves between PNM and MIO were nearly identical. Tranter et al. [26] also showed very similar intrusion results between PNM and MIO. They also showed that the intrusion curves shifted when the sample images were compressed, and the MIO result showed better agreement with the experimental curve than the PNM result. However, neither of them performed transport simulations with the saturated images obtained from the MIO technique.

Sabharwal et al. [30] applied FM to investigate the liquid water intrusion in dry GDL and compared them to the actual saturation images. A comparison of the water distributions at similar saturation showed good agreement between μCT and FM results. A major difference, however, was the appearance of smaller and discrete liquid water clusters compared to the larger water clusters in the μCT reconstructions. A comparison of dry and wet relative diffusivity obtained from both μCT and FM images showed great agreement to each other in both the in-plane and through-plane directions. All those studies show that the FM approach provides a good prediction of a non-wetting phase intrusion in the PEMFC and PEWE porous media with sufficient accuracy. However, they did not perform the permeability simulations.

The major gap observed was that several authors have employed either PNM or DNS to predict transport properties of porous media in PEMFC and PEWE. Few literature were found using both approaches, and compared the results to each other and to the experimental data. In this thesis, the FM model is used to generate virtual saturation images of liquid water at desired saturation levels and the transport properties are computed using a finite element based DNS model. Both the PNM and the DNS approaches are used to study the difference of the two numerical approaches in the PEMFC and PEWE porous media microstructure modelling.

1.3.2 Experimental characterization

Porosity and pore size distribution are the two most common parameters used to characterize a porous media. The estimation of these parameters can be achieved both numerically and experimentally. Direct imaging followed by image analysis techniques, such as image segmentation and filtration, have been used widely for numerical characterization of porous media. A widely used experimental method for estimating the PSD is a mercury intrusion porosimetry (MIP). A brief explanation of the techniques used in this thesis is given in this section.

1.3.2.1 Imaging

Tomographic imaging techniques have been applied for analyzing fuel cell porous media by visualizing the microstructure of the media. In particular, micro and nanoscale X-ray computed tomography (μCT and nano-CT), and focused ion beam-scanning electron microscopy (FIBSEM) have been used for visualizing the micro- and nanostructure of porous materials.

X-ray computed tomography offers both micro and nanoscale visualization and it is non-destructive. For these reasons, this method is one of the most widely used techniques. The computed tomography works by taking many 2D radiographs of a sample while the sample is rotated 180° or 360°. From these raw images, a 3D image stack can be reconstructed, which allows characterizing microstructure of the porous media. μCT can achieve voxel resolution of around 1-5 μ m which enable the distinction of the smallest size pores in fibrous PTLs without MPL. μCT has been used for visualizing single and dual layer PTLs [32, 57–59]. The achievable voxel resolutions of nano-CT (nCT) is about 50 nm [60]. nCT has been used for imaging nanoscale porous media, such as catalyst layers (CLs) and micro porous layer [60–62]. Additionally, μCT has been used to obtain tomography images of partially saturated PTLs [31, 63, 64].

FIBSEM uses an ion beam to slice the sample and an electron beam to image the exposed sample. For this reason, this is a destructive imaging technique. However, the resolution it can achieve is about 2 nm/pixel which is sufficient to capture nanoscale pores of CLs [13, 65–70]. However, the resolution in the slicing direction is 10-20 nm, which can lead to loss of information in the same direction [69].

There are two methods commonly used to visualize water behaviour in an operating PEM fuel cell: neutron imaging [13, 16, 71, 72], and synchrotron X-ray imaging [31, 57, 73, 74]. In neutron imaging, the different decay of the neutron beam intensity through the sample and water is used to identify the regions with liquid water. The neutron beam has a high sensitivity to molecules with low atomic numbers but low sensitivity to metals. This characteristics allow capturing the amount of liquid water in the cell. The spatial resolution of the neutron imaging is about 10 μm [75].

Synchrotron X-ray imaging is capable of resolving the liquid water distribution in the PTL with a temporal resolution less than a few seconds. The spatial resolution of the synchrotron imaging is about 1 μm. The X-ray beam interacts with the sample as it is transmitted. The attenuated intensity of the X-ray beam at each location depends on the material and the thickness. Since the X-ray beam is highly attenuated when travelling across high atomic number materials, the metal parts have to be replaced for accurate imaging of water distribution. Manke et al. [76] visualized the eruptive water emergence of liquid water into the channel from the GDL. Haußmann et al. [73] investigated water transport in GDL and showed that cracks in the micro porous layer can serve as water transport pathways to the channel. Chenvalier et al. [72] tested the precision and accuracy of liquid water measurements in operating PEM fuel cells. They proved that the accuracy of the synchrotron X-ray radiography can be impacted by the noise level of the CCD camera.

In this thesis, μCT is used to obtain tomography images of the PTL materials. The choice of μCT is based on the characteristic pore size of the materials used in this thesis. In-situ experiment to analyze the water distribution of the material could be used to further validate the material results. However, such experiments require access to synchrotron facilities and are challenging and time consuming. Therefor, they are not within the scope of this thesis.

1.3.2.2 Mercury Intrusion Porosimetry

Mercury intrusion porosimetry (MIP) is an analytical method for the determination of the cumulative intrusion pore volume of porous media. Mercury is forced into the void part of the material by increasing its pressure and the mercury intrusion is recorded at each liquid pressure. The relation between the applied pressure and amount of intruded mercury is used to determine total pore volume and pore size distribution (PSD). To obtain PSD, some assumptions are made: a) contact angle between mercury and the surface solid is constant and the surface tension of the working fluid is known. b) the shape of the pores are assumed cylindrical for estimating the pore radius based on the invasion pressure.

One of the main advantages of mercury is that it is non-wetting to almost all materials. Using mercury as the working fluid is beneficial to obtain the overall PSD of the media. However, differentiating between hydrophobic and hydrophilic networks is not possible since mercury is highly non-wetting. The capillary pressure curves using MIP will be used to assess the capillary pressure curves obtained numerically.

1.3.2.3 Transport Property Estimation

Mass transport in fuel cells and electrolyzers is a key process because poor mass transport leads to significant performance losses. Mass transport occurs mainly by convection and diffusion. Convection refers to the transport of a species by bulk motion of a fluid under the action of a mechanical force. Diffusion refers to the transport of a species due to a gradient in concentration.

1.3.2.3.1 Diffusivity Many experimental studies have been carried out on various PTL samples to measure the effective diffusivity. LaManna et al. [77] studied the effect of MPL, GDL thickness and PTFE in through-plane direction for three different types of GDLs. They found that the effective diffusion coefficient decreased linearly with increasing PTFE while the introduction of an MPL reduced the coefficient by 26 - 38%. Thickness had no effect on single layer GDLs but lower thickness resulted in low diffusion coefficients for MPL coated GDLs.

Hwang et al. [2] applied an electrochemical limiting-current method to measure the effective diffusivity in the through-plane direction for uncompressed, dry and wet samples. They found that the porosity is the critical factor that controls the effective diffusivity and increasing loading reduces the porosity and the available diffusion pathway. The PTFE treated materials showed better gas diffusion because the hydrophobic PTFE pathways.

Unsworth et al. [78] investigated effective diffusion coefficients and the effect of thickness and microstructure of Toray and SolviCore GDLs using the closed-tube method. They used a Loschmidt cell which is an experimental apparatus using the closed-tube method. They found that diffusibility (effective diffusivity normalized by the bulk diffusivity) varied with thickness. They compared the diffusivility of the Toray GDLs (Toray 060, and 120), which have a uniform microstructure but different thickness. The difference of the diffusibility between two samples were 8%. They then proposed a modified correlation that relates diffusibility to porosity for carbon paper GDLs with less than 10%-wt. of PTFE.

Mangal et al. [79] used a diffusion bridge to measure the through-plane diffusivity of Toray gas diffusion layers. Two plates with a gas channel were assembled together and the PTL was placed between the two plates. Compressed oxygen and nitrogen gases were sent through each side of the gas channel and both gases were controlled by a mass flow controller for uniform flow rate. A differential pressure controller was used to control the static pressure difference between the gas channels. A back pressure controller was connected to the nitrogen channel and used to control the static pressure of the system. They tested same PTLs with different PTFE loading and found that the diffusivity of the PTLs were decreasing with increasing PTFE loading.

Rashpov et al. [80] measured in-plane effective diffusion coefficients of Toray, SGL, and Freudenberg GDLs as a function of compression. They found the presence of carbonaceous binder have significant impact on the effective diffusivity. Furthermore, compression of the porous samples led to more tortuosity. Especially, samples with high PTFE contents showed becoming more tortuous when they were compressed than the samples with low PTFE.

Xu [81] used a diffusion bridge for measuring in-plane diffusivity. Various PTL and CL samples were tested with different compression levels. The diffusion bridge setup used by Mangal et al. [79] was enhanced to achieve a better accuracy. They found that the GDL materials with less carbon matrix and large porosity showed the highest diffusibility, i.e., SGL 29BA and Freudenberg H2315. Additionally, when the results were compared with the theoretical models, the error increased with a lower porosity and a higher compression level. They proposed that the Knudsen effect should be considered at a high compression level.

In this thesis, the experimental dry gas diffusivities in the through-plane and in-plane directions measured using the diffusion bridge setup proposed by Mangal et al. [79] and Xu [81] are compared to the values obtained by numerical simulations.

1.3.2.3.2 Permeability Gostick et al.[82] measured the gas permeability of several common PTLs in three perpendicular directions and compared them with the analytical models . The data were well described by the Carman-Kozeny model as a function of porosity and Carmen-Kozeny constants were determined for each material. Additionally, they compared the data to the Tomadakis-Sotirchos (TS) model and found to agree well. Since the TS model requires no fitting parameters, it was appropriate for predicting anisotropic materials.

Koido et al. [83] measured the relative gas permeability of Toray 060 material with 5% PTFE contents using steady-state method. A GDL test piece was sandwiched between similar GDLs. The GDL at the inlet was set to ensure homogeneous distribution of water within the GDL test piece and the outlet GDL was for minimizing the effect of the outflow boundary. The measured gas permeabilities showed decreasing permeability with increasing saturation. They also computed relative permeabilities with computational modeling. The experiments and the modeling showed similar decreasing trends for gas relative permeability.

Hussaini et al. [84] experimentally measured absolute permeability and air-water relative permeability for fuel cell GDL materials such as Toray carbon paper (060, 090, 120) and E-Tek carbon cloth. They found that the carbon paper materials showed 18% higher in-plane absolute permeability than the through-plane permeability. They suggested that measuring the in-plane permeability showed much less uncertainty than in the through-plane direction.

Mangal et al. [79] tested through-plane permeability of Toray GDLs using the same diffusion bridge. The permeability of Toray samples was found to be the range of $1.13 \times 10^{-11} - 0.35 \times 10^{-11}$ m^2. They compared the experimental data to Carman-Kozeny equation and the prediction of permeability showed high accuracy for low PTFE content samples but discrepancies occurred at high PTFE samples.

Xu [81] measured in-plane permeability of various GDLs and compared the experimental results to the Carman-Kozeny model and found that the SGL-BA series materials showed increasing of error with increasing volume fraction in matrix. They concluded that the reliability of the Carman-Kozeny model dependent on the maximum pore diameter and the amount of carbon matrix.

In this thesis, experimental dry gas permeabilities of GDL, PTL, and CL samples were used to validate modelling predictions. The same diffusion bridge mentioned in Section 1.3.2.3.1 was used to measure the gas permeability of each sample with a variation, which a single gas is forced to pass through the PTL samples.

1.4 Objectives

The objectives of this thesis is to study the validity of PNM and DNS approaches to estimate transport properties based on μCT images. Based on the literature review in these areas, the following research gaps are identified:

- Two popular modelling approaches, i.e., PNM and DNS, have been used in many studies. However, none of these studies have employed both of these methods to estimate transport properties of GDLs, electrolyzer PTL and CL samples.

- Many studies have used pore size distributions (PSDs) for characterizing the microstructure. Several modelling studies have used PSDs to validate their microstructure models, but only few of those studies compared the numerical PSDs with the experimental PSDs.

- There have been many studies about predicting transport properties of fuel cell GDL materials. However, analyzing transport properties of PTLs for proton exchange water electrolyzer (PEWE), such as Titanium powder based material, has not been well described either experimentally or computationally.

Based on the aforementioned gaps, the objectives of this thesis are:

- Create tomography images of morphologically different porous media using μCT.
- Simulate liquid water and mercury intrusion in porous media to compare simulation results from PNM and DNS.
- Use tomography images to predict dry permeability and diffusivity of the porous media with the two numerical approaches and evaluate them by comparing to experimental properties.
- Estimate relative transport properties and compare the results between the PNM and DNS approaches.

Chapter 2

Methodology

As discussed in Chapter 1, one of the goals of this thesis is to use numerical simulation to estimate transport properties of PEMFC and PEWE porous media from X-ray CT tomography images. This chapter describes the theory and algorithms used for image acquisition, analysis, and statistical functions. The PNM and DNS methods used to simulate gas and water transport, as well as liquid injection, are also described.

2.1 Imaging

Many different methods have been used for visualizing the microstructure of porous media. X-ray micro computed tomography (μCT) is a non-destructive imaging technique that is widely used to characterize PEMFCs gas diffusion layers. μCT is a powerful method to investigate structures in the 1 μm to 25 μm range. Another popular methods for visualizing the nanoscale structures of porous media, such as fuel cell catalyst layers [32, 69], are focused ion beam-scanning electron microscopy (FIBSEM) and nano-CT. In this work, μCT was selected for image acquisition as the focus of this work was analyzing the microstructure of GDLs and PTLs.

Figure 2.1 shows a flow chart of the image processing procedure followed to generate a binarized image for μCT data. As described before, μCT is commonly used to obtain three dimensional image volumes with microstructural information of the sample. This is done through stacking of many two dimensional radiographs taken at different angles and performing reconstruction using a back projection algorithm [85] to generate a three dimensional sample stack. Reconstruction of the raw images generates a set of cross sectional images from the radiographs known as the raw reconstruction stack. The three dimensional stack contains microstructural information as a function of the X-ray absorption coefficient of the material at each location. The raw reconstruction stack is then aligned and cropped based on a selected re-

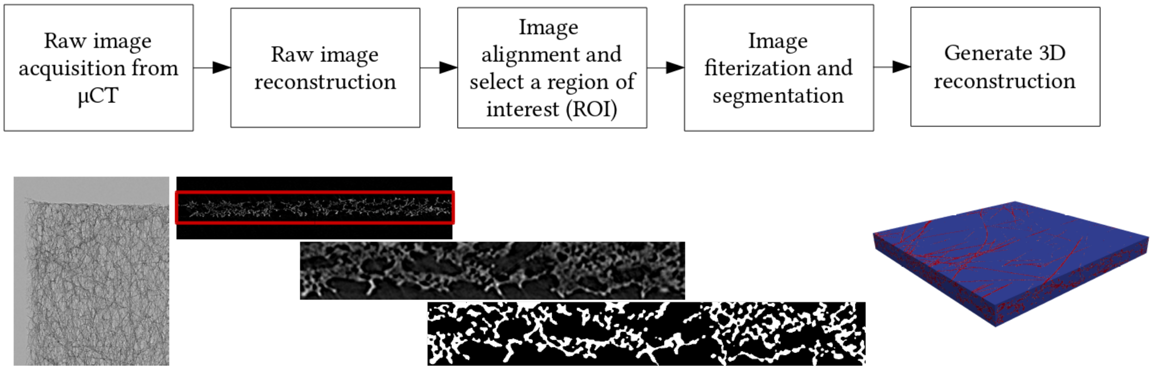


Figure 2.1 – Schematic of the image processing steps for μCT data.

gion of interest (ROI). Alignment is required to place each tomography image at the same position. The reconstruction stack is then filtered and binarized to segment the grayscale images into binary pore and solid regions. The following sections describe each image acquisition and processing steps, in detail.

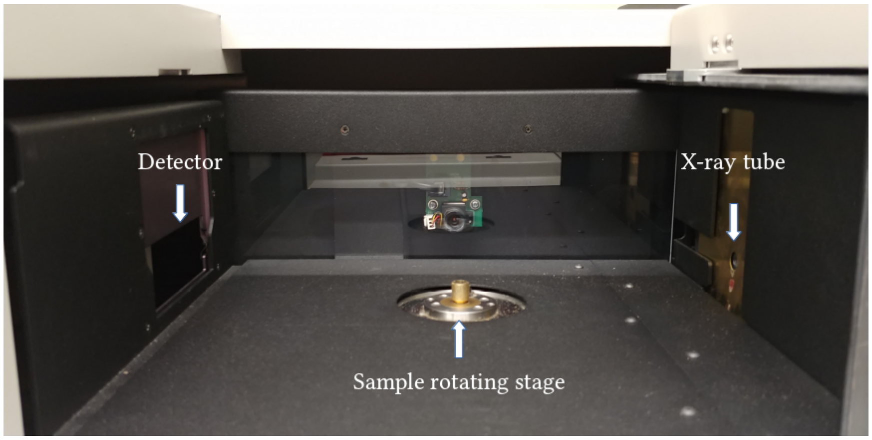
2.1.1 Image acquisition

Obtaining a three dimensional image using μCT requires combining 2D radiographs acquired at a specific angle range. That means the sample is rotated in small steps in a range of either 180° or 360°. μCT imaging is based on the attenuation difference of X-rays depending on the elements of the sample. The resulting scanned image shows the contrast between the phases of the material. The X-ray attenuation is governed by:

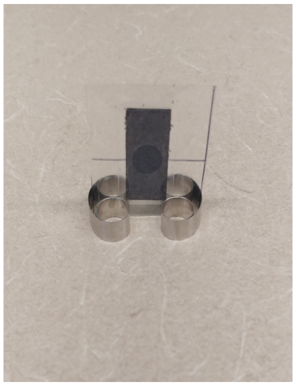
$$I = I_0 e^{-\mu t} \tag{2.1}$$

where I is the intensity of the X-rays after passing through the thickness t, I_0 is the incident X-ray intensity, and μ is the linear attenuation coefficient of the material. The detected X-ray intensity, I, is a measure of the attenuation coefficient based on the sample position.

Image scanning was performed using a Bruker SkyScan1172 micro CT located at the Centennial Centre for Interdisciplinary Science (CCIS), University of Alberta. The sample should be centred and orthogonal to the sample rotation plate to get accurate 2D tomograph results of the sample. Furthermore, the sample should be attached onto the rotation plate which is held to the rotation stage. In the case of PTL samples, they are too thin to be placed perpendicular to the sample rotation plate. For this reason, a sample holder is required to place the sample in an orthogonal manner. Two stainless steel sample clips are used for the sample holder to hold the



(a)



(b)

Figure 2.2 – (a) Internal view of Bruker SkyScan 1172. (b) Sample holder used to hold the sample on the rotation stage.

sample vertically. Figure 2.2 shows an internal view of the μCT equipment and the sample holder design.

As can be seen in Figure 2.3, magnification of the sample depends on the position of the sample. The sample has the greatest magnification when it is placed close to the x-ray source. The resolution of a CT scanner is represented as the voxel size which can be determined by the magnification (M) and the distance between pixels. The magnification is determined as follows [86]:

$$M = \frac{A + B}{A} \tag{2.2}$$

where A is the distance from the x-ray source to the sample, and B is the distance from the sample to the detector. The maximum magnification can be achieved by positioning the sample as close to the detector as necessary to fit the projection within the field of view of the detector. In order to separate solid from pore space achieving good contrast between phases is of paramount importance. There are three major

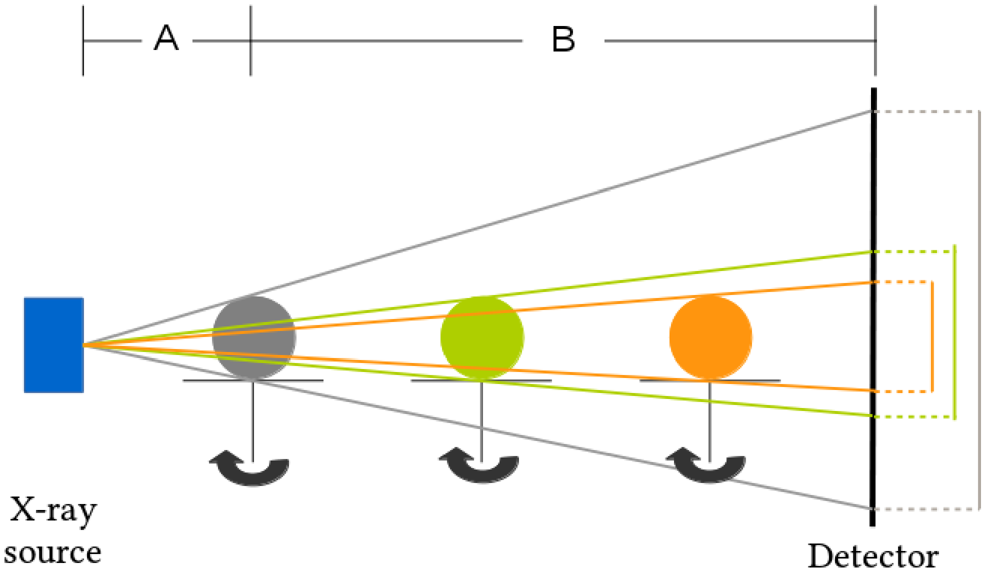


Figure 2.3 – Schematic setup of μCT showing objects in three different positions

parameters required to be optimized to achieve optimal contrast between phases: i) X-ray energy, ii) voxel size, and iii) exposure time.

The X-ray energy determines the transmission rate of the X-rays through the sample, and as a result it determines the contrast of the resulting images. Too high energy would result in images that are entirely white, while too little energy would not allow any X-rays to transmit across the material. The source voltage, which controls the energy intensity, has therefore a significant impact on the final result depending on the material. The optimal source voltage highly depends on the material composition of the sample. A material with low atomic number, such as carbon fiber paper, requires low X-ray energy to achieve high contrast because those materials attenuate less X-rays than high atomic number material [87]. In this work, for the carbon fiber material, a 29 kV source voltage was used since the density and thickness of the carbon paper are small. On the other hand, a source voltage of 90 kV was used for imaging Ti material as Ti has a higher atomic number than carbon [88].

The voxel size refers to the length of each voxel which determines the size of reconstructed 3D data. SkyScan 1172 provides maximum magnification of 1 μm, but a small voxel size leads to longer acquisition times. Scanning with low resolution relative to the size of the structure of interest can cause an underestimation of the density owing to an overestimation of object thickness [86]. The ratio of voxels to the

Table 2.1 – μCT operational parameters

Parameter	**Ti**	**SGL 39BA**
Source voltage (kV)	90	29
Current (μA)	112	175
Rotation step(°)	0.2	0.2
Random movement	10	10
Averaging	3	3
Resolution (μm)	1.79	1.79
Duration	6 h	5 h
Exposure (ms)	5301	3534
Filtration	Al 0.5 mm	–

object size should be higher for accurate morphologic measurements. The exposure time is determined automatically by the software based on selected X-ray energy, voxel size, and the parameters mentioned in Table 2.1. The exposure time refers to the time to record one radiograph.

In Table 2.1, the parameters used for μCT scan are specified. Averaging option is the number of images used to average the shadow projection. A higher number for frame averaging will increase the signal to noise ratio. Random movement sets the number of pixels for random synchronous movement of acquisition area. This can be activated to reduce ring-artifacts produced from local irregularities of sensitivity of the sensor. Based on these parameters, the total duration can be determined as follows [89]:

$$T = N \times (P + (E \times A)) \tag{2.3}$$

where T is total duration, N is the number of radiographic images, P is a pausing parameter, E is the exposure time, and A is the averaging parameter. The pausing parameter, P, can be used to remove afterimage effects. Residual information from the previous image could appear on the next image when the scanning is too fast.

Additionally, filtration can be added when objects are dense. Filters reduce beam hardening artifacts by absorbing low energy X-ray. The filter is placed between the X-ray source and the sample. Frequently used filters are 0.1 to 2 mm of copper and 0.5 to 1.5 mm of tin, combinations of both, and aluminum [87]. Beam hardening is the most common CT artifact, which appears when the X-ray beam encounters differences in absorption from different angles and along different paths through the object [87].

2.1.2 Image reconstruction

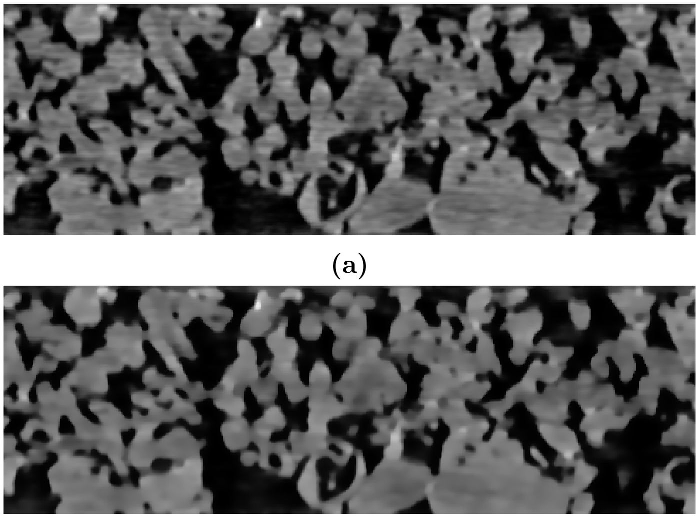
After all 2D radiographs are obtained, a set of grey-scale cross sectional images of the material can be reconstructed to create a 3D stack. This was done with the NRecon software (Bruker corp.). The algorithm used for the reconstruction is the Feldkamp filtered back-projection algorithm implemented in the NRecon software [85].

When the radiographs are transformed to a 3D stack, they often contain noise which may be attributed to the scanning steps. There could be artifacts in each 2D reconstructed image, caused by variations in the sensitivity of the detector, which could be an afterimage. These sort of artifacts are usually removed by applying a flat-field correction technique which removes the effect of pixel-to-pixel sensitivity variation across the array. Other commonly occurring artifacts are ring artifacts and beam hardening artifacts. Ring artifacts are caused by a miscalibration or defective detector elements, which results in rings centered on the center of rotation of the 2D tomography images. To reduce ring artifacts, ring artifact reduction is used. If the cause of the artifact is the miscalibration, it can be reduced by rescanning the object after a calibration. If it is due to a defective detector, the ring artifact reduction function in NRecon mitigates the artifact using filtering methods [86]. Beam hardening causes the edges of an object to appear brighter than the center, even if the material is the same throughout. This artifact occurs when penetrated X-rays are insufficient. The method to reduce this artifact is called beam hardening correction. The reconstruction software NRecon provides methods for reducing both artifacts. However, there could still be remaining noise in the reconstructed images. More detail discussion about the CT artifacts can be found in reference [90, 91].

2.1.3 Filtering and binarization

Due to the remaining noise, further noise reduction is required. In order to remove noise from the reconstructed set of μCT images, a bilateral filter is applied on the images [92]. The filter smooths images while preserving edges. The filter replaces the intensity of each pixel with an average intensity of nearby pixels. This leads the pixels in the same phase to have similar intensities as shown in Figure 2.4. It can be seen that the filtered image shown in Figure 2.4b has less noise than Figure 2.4a.

The filtered images can then be binarized. Image binarization is the process of taking a grayscale CT image and converting it to black and white. When it comes to the pore scale PTL modeling, the binarization is crucial since it determines the porosity and microstructure. The reconstructed set of images were segmented to binary form using the FIJI/ImageJ software packages [93]. Image segmentation can



(a)

(b)

Figure 2.4 – μCT image at different preprocessing stages, a) raw reconstructed image aligned and cropped, b) image after applying 3D bilateral filter.

be done using either thresholding algorithms provided by FIJI or manually selecting a pixel intensity that divides two phases into the black and white regions based on the grayscale intensity histogram. In this work, the conventional thresholding algorithms implemented in FIJI were tested. Finally, Otsu's and Sauvola's algorithms are used for the fibrous samples and Ti, respectively. These algorithms are chosen since they were able to generate binary images with similar porosity to experimental data.

For Ti samples, a Sauvola algorithm was applied for segmenting filtered images [94]. Sauvola's method computes a local threshold for each pixel individually taking into account image intensities in the local neighbourhood of the pixels. The threshold, $t(x, y, z)$, is computed using the mean, $m(x, y, z)$, and standard deviation, $s(x, y, z)$, of the pixel intensities in a selected window:

$$t(x, y, z) = m(x, y, z)\left[1 + k\left(\frac{s(x, y, z)}{R} - 1\right)\right] \tag{2.4}$$

where R is the maximum value of the standard deviation ($R = 128$ for a grayscale image), and k is a parameter which takes positive values in the range [0, 1]. Parameter k and window size need to be selected by trial and error based on the visual comparison. Selected parameters in this work were 20 for window size and 0.5 for the k value.

For fibrous carbon samples, a global thresholding Otsu algorithm was applied [95]. The algorithm presumes that the image has a bi-modal intensity histogram of two phases. Otsu's thresholding method iterates through all potential threshold values to determine the best threshold value that divides the foreground and background. The threshold value can be defined as a weighted sum of variances of the two phases:

$$\sigma_w^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t) \tag{2.5}$$

where weights w_0 and w_1 are the probabilities of two phases with respect to the entire pixels, t is a selected threshold value, and σ_0 and σ_1 are variances of the two phases. As the goal of this algorithm is to find the threshold value that minimizes the variances between two phases, the threshold value with the least σ_w^2 is selected. After the binarization, some imperfections could happen due to the fact that the grayscale intensities of carbon matrix are relatively in lower range than those of the carbon fibers. This can cause that some portion of carbon matrix are not segmented as solid. In order to fill the imperfections, a 3D morphological closing and opening with a spherical structural element of 3 voxels was applied to the 39BA samples.

The initially generated 3D image stack is aligned and cropped to $1500 \times 1500 \times \delta_t$ voxels (where δ_t is the thickness of each image stack).

2.1.4 Stochastic reconstructions

Stochastic reconstructions provide an alternative to direct imaging techniques to generate porous media. Using this method porous media with different morphology can easily be generated in order to understand the effect of varying morphological factors such as fiber diameter. Furthermore, for nanoscale media, such as CL, stochastic reconstructions can be a good option, since reconstructions using FIBSEM and nano-CT are time consuming and might contain artifacts.

In this work, stochastic reconstructions for CL samples were generated using a random overlapping sphere based algorithm. The algorithm has been implemented in pyFCST by Sabharwal et al. [96]. More details about the algorithm can be found in reference [54]. The algorithm inserts spheres in an empty domain for the solid phase. The size of spheres can be defined as a constant radius value or a distribution. To pack spheres in the domain, a random location is selected to be used as a center. As the spheres are placed in the domain, they might overlap one another. The amount of overlapping can be controlled by adjusting penetration parameter, ψ. Spheres keep being placed in the domain until the desired porosity is achieved.

2.1.5 Statistical functions

Statistical functions provide information about the probabilities of geometric features. These functions can be used for characterizing heterogeneous porous media because each statistical function provides information about different aspects of the porous media. There have been many studies proposing a number of statistical functions for porous media [97–100]. In this thesis, the two-point correlation and chord length functions are used. The two-point correlation function provides the probability of finding any two points at a certain distance, r, in the same phase, i, which may be used to estimate the phase volume fraction and interfacial area. The chord length function is the probability of finding a chord of given size in one phase, which may be used to estimate the size distribution of the phase. A chord is the line segment between immediate inter-phase boundaries. This section describes the two-point correlation function and chord length function which are used for porous media characterization in this work.

2.1.5.1 Two-point correlation function

The two-point correlation function, $S_2^{(j)}(\overrightarrow{\mathbf{r_1}}, \overrightarrow{\mathbf{r_2}})$, is the probability of finding any two points, $\overrightarrow{\mathbf{r_1}}$ and $\overrightarrow{\mathbf{r_2}}$, in one phase j [98]. This can be calculated by counting the number of times that both ends of a line of length r fall in the same phase. This number is then normalized by the total number of translations to obtain the probability. The two-point correlation function can be defined as [98]:

$$S_2^{(j)}(\overrightarrow{\mathbf{r_1}}, \overrightarrow{\mathbf{r_2}}) = P\{I^{(j)}(\overrightarrow{\mathbf{r_1}}) = 1 \quad \text{and} \quad I^{(j)}(\overrightarrow{\mathbf{r_2}}) = 1\} \tag{2.6}$$

where $\overrightarrow{\mathbf{r_1}}$ and $\overrightarrow{\mathbf{r_2}}$ are two arbitrary points in the phase j, and the characteristic function $I^{(j)}(\overrightarrow{\mathbf{r}})$ is defined as:

$$I^{(j)}(\overrightarrow{\mathbf{r}}) = \begin{cases} 1, & \text{when } \overrightarrow{\mathbf{r}} \text{ is in phase } j. \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

Figure 2.5 provides an illustrative example for obtaining two-point correlations.

A porous media can be assumed as statistically homogeneous, if the averaged value of probability functions are similar in all subsamples of the physical space [100] and if its probability distribution functions are the same for a transverse shift [99].

For statistically homogeneous media, the two-point correlation function depends only on the distance between two points, $r = ||\overrightarrow{\mathbf{r_1}} - \overrightarrow{\mathbf{r_2}}||$, which can be expressed as $S_2^{(j)}(r)$. This leads to following relations [98]:

$$S_2^{(j)}(0) = \phi_j \quad \text{and} \quad \lim_{r\to\infty} S_2^{(j)}(r) = \phi_j^2 \tag{2.8}$$

where ϕ_j is the volume fraction of phases j.

Once the correlation function of one phase is known, the other functions which are found in different phases can be obtained easily since they are linearly dependent.

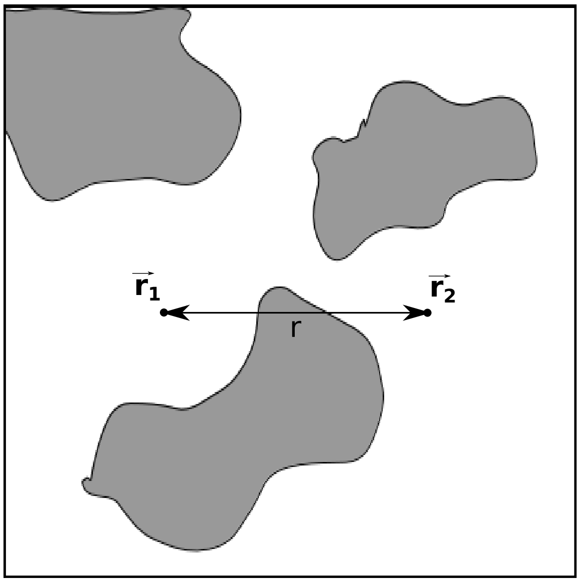


Figure 2.5 – A schematic diagram of two-point correlation function estimation.

This feature of the two-point correlation function can lead to an estimation of interfacial area between two phases. The specific surface area of a two-phase medium can be obtained as a function of its two-point correlation function as follows [100]:

$$s_j = -\beta \frac{d}{dr} S_2^{(j)}(r) \bigg|_{r=0} \tag{2.9}$$

where s_j is the specific inter-phase area, and β is 4 for 2D and 6 for 3D. The specific surface area can be used to estimate permeability of the porous media by using the Kozeny-Carman relations.

2.1.5.2 Chord length function

The chord length function, $C^j(r)dr$, is defined as the probability of finding a chord of length between r and $r + dr$ in a phase j [101]. Chords are defined as the lengths between intersections of line with the two-phase interface [97, 101]. In order to calculate the chord length function, all the chords in a given direction are recorded and binned by their frequency. The chord length function is then obtained by normalizing computed chords by the total number of chords. Figure 2.6 shows an illustration of void chords in a two-phase image.

The mean chord length (λ_D) is the first moment of the chord length function defined as [101]:

$$\lambda_D = \int_0^\infty rC(r)dr \tag{2.10}$$

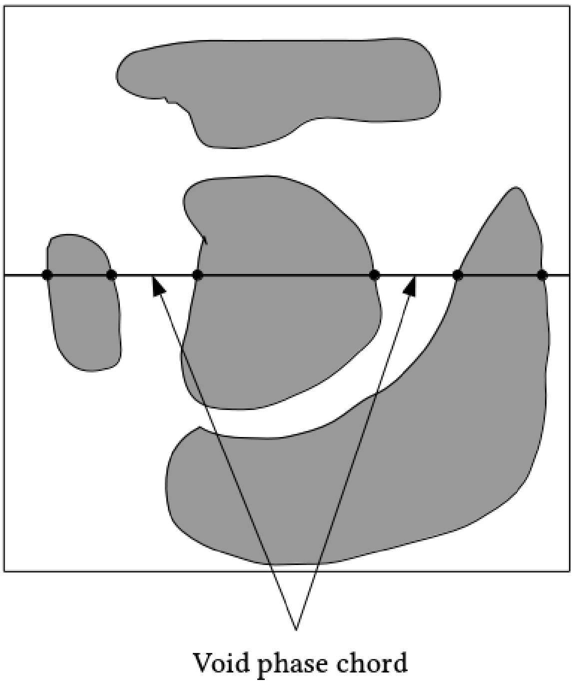


Figure 2.6 – A schematic diagram of chord length function estimation.

The chord length function contains information of clusters in the image so that it includes the implication of pore size distribution function.

The statistical functions such as two-point correlation and chord length functions are implemented by Pant [100] in the Porous Media Stochastic Reconstruction Toolbox (PMSRT) and it is used for this thesis.

2.2 Pore network modeling

Pore network modeling (PNM) is a popular approach for simulating transport in porous media. PNM solves 1D analytical solutions of the relevant transport equations on a simplified pore space. The pore space is described as a network of pores and throats and some idealized geometries, such as spherical pores and cylindrical throats, are assigned for each pore and throat. Generating pore networks and running transport simulations are done using OpenPNM: A Pore Network Modeling Package [36].

A pore network can be either structured or unstructured. The structured model is a network constructed based on a rigid lattice such as a cubic lattice. Each nodal point is assigned a pore location, and the connection between two nodal points becomes a throat. Pore and throat sizes can be assigned randomly or using statistical distribution functions. Representing realistic PTLs with structured networks is challenging since it is difficult to express same feature of the PTL microstructure, such as entangled fibers, within a cubic network. In unstructured networks, nodal point locations and pore connections are not predetermined. The location can be assigned with any of the following methods: a) randomly distributed nodal points [102], b) using specific tessellation functions, such as Voronoi and Dellaunai tessellations [42], c) analyzing images obtained from either stochastic reconstructions, or d) μCT images using a watershed partitioning algorithm [43]. This technique is termed network extraction. In this work, unstructured networks obtained using the watershed partitioning network extraction algorithm in reference [43] on either stochastic reconstructions or μCT images.

Once the network is extracted, the pore size distribution can be obtained directly, since pore and throat information was determined during the network extraction. Based on the size information, any simulation can be performed in PNM, such as mercury and liquid water intrusion, and dry and wet transport simulations. This section provides the theoretical background of all these simulations in PNM.

2.2.1 Network extraction

Pore network extraction can be used to create a pore network based on a stack of segmented μCT images. The network extraction is done using the open source porous media image analysis toolkit, PoreSpy [103]. The source code for the network extraction is shown in Appendix A. The algorithm uses a watershed segmentation to determine the pore region based on the distance from the solid region. The SNOW algorithm proceeds in the following steps [43]:

1. Obtain Euclidean distance map of the pore space
2. Apply a Gaussian filter on the distance map to reduce the occurrence of plateaus
3. Apply a maximum filter with a spherical structuring element of radius R on the smoothed distance map to identify peaks in the image
4. Trim saddle points and merge nearby peaks
5. Obtain separate pore space using a watershed segmentation
6. Obtain pore network information based on local peaks of segmented regions as nodal points.

The Euclidean distance between each pore voxel and the nearest solid voxel is calculated using the scipy image module. As the distance map of the pore region contains many spurious peaks (local maxima), applying the Gaussian filter with sigma of 0.4 removes some of the erroneous peaks, which is shown in Figure 2.7b. The maximum filter with a structural element of $R = 4$ is then applied. The maximum filter replaces the value of each voxel with the lightest voxel found within its neighbours, defined by structural element [43]. The maximum filtered image is shown in Figure 2.7c. The intensity values in the distance map and the maximum filtered image are compared to each other. This enables identifying peaks of the image by storing the same values between the maximum filtered distance map and the previously smoothed distance map. The selected peaks are shown in Figure 2.7d.

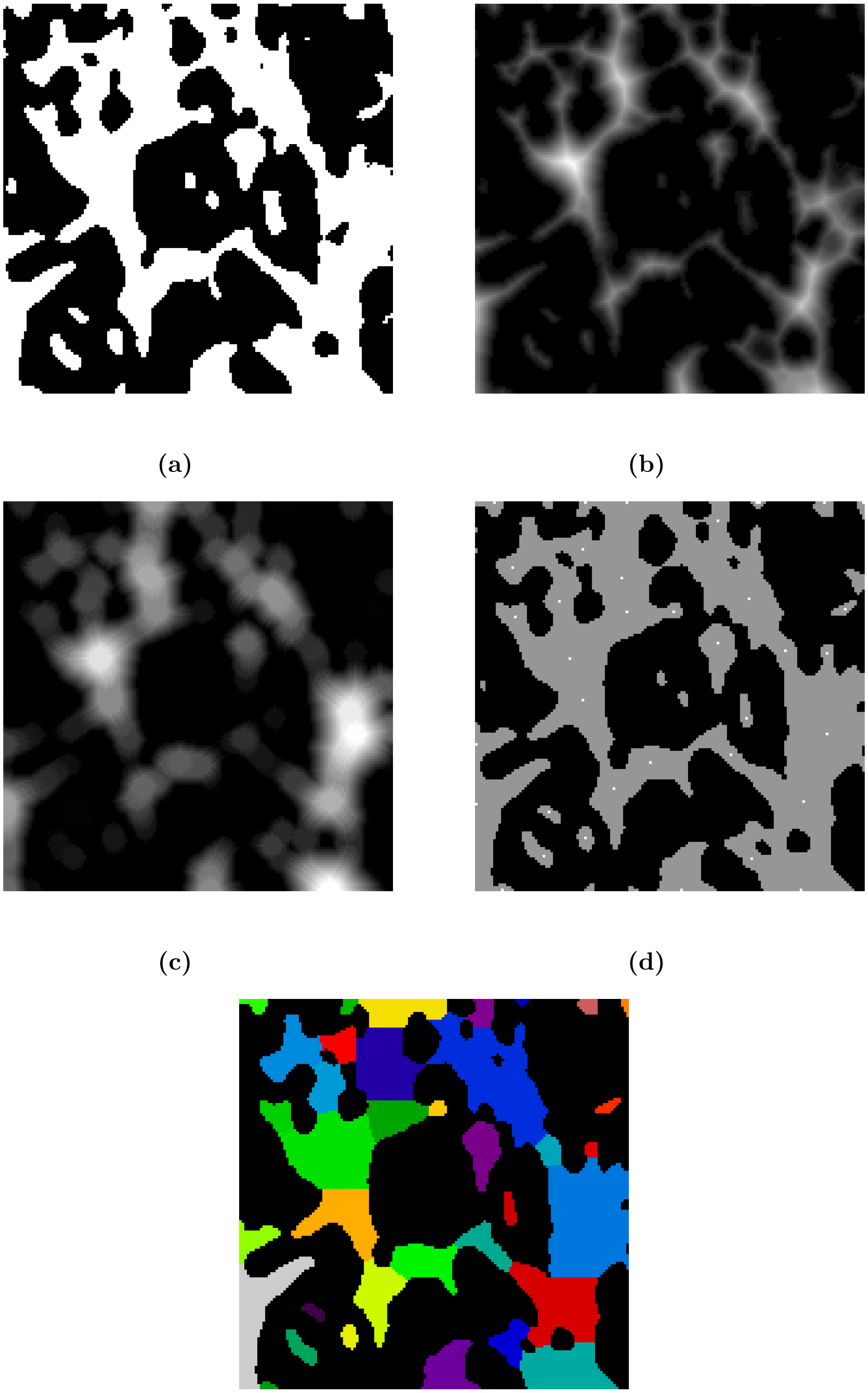
Once a set of peaks are found, peaks on saddles points are removed. They were identified as peaks erroneously in the previous step, but they need to be removed since they are connected to voxels with higher values. Removal of these peaks eliminates thin regions between solids [43].

The next step is merging peaks placed nearby, which would occur in large pore spaces. At the point when the sets of peaks are discovered that are nearer to one another than the solid, the one more far from the solid is kept. This behaviour will avoid generating two overlapping pores in one large pore space which will change the size distribution of the network [43].

The peaks arranged from the above steps are used as markers for the watershed segmentation and nodal points to create spherical pores of local pore spaces. A marker based watershed algorithm is used to segment the pore region as shown in Figure 2.7e. The extracted network includes pore location, connectivity, and pore and throat size information. The size data include two different cases, which are inscribed diameter and equivalent diameter. The inscribed diameter is calculated from the diameter of the sphere fitted in a segmented void region whereas the equivalent diameter is computed from the sphere with the same volume as the segmented local pore region. The information obtained from network extraction can be directly used for calculating pore and throat size distributions and running transport simulations in the open source pore network modeling framework, OpenPNM [36, 104]. More details about the algorithm can be found in reference [43].

2.2.2 Intrusion

To study multiphase flow in porous media using PNM, a quasi-static intrusion algorithm for the pore network is required. When referring to transport of a non-wetting



(a) (b)

(c) (d)

(e)

Figure 2.7 – Network extraction steps using a watershed segmentation in PNM: a) A binary image to be extracted, b) Euclidean distance map of the pore space, c) The distance map applied by a maximum filter d) Peaks obtained by comparing the distance map and the maximum filtered image, and e) a result image of the watershed segmentation.

fluid in porous media, intrusion is defined as the displacement of the wetting phase by invasion of the non-wetting phase. There are two algorithms addressing the invasion of non-wetting phase into porous media implemented in OpenPNM: a) Ordinary percolation (OP), and b) Invasion percolation (IP). The capillary pressure of each pore is calculated based on the throat diameters using the Washburn equation:

$$p_c = -\frac{2\gamma cos\theta}{r_p} \tag{2.11}$$

where γ is the surface tension of the liquid, θ is the contact angle between the solid surface and the liquid, and r_p is the pore radius. The capillary pressure p_c is defined as :

$$p_c = p_l - p_g \tag{2.12}$$

where p_l and p_g are liquid and gas pressure. It is assumed that p_g is zero. In this section, the two intrusion algorithms are explained, which are used for simulating mercury and water intrusion to obtain microstructural information and results of two phase simulation.

2.2.2.1 Ordinary percolation

For mercury intrusion porosimetry (MIP), the ordinary percolation (OP) algorithm is used. Appendix A provides an example of the MIP code. All six faces of the network are assigned as boundaries. The OP algorithm proceeds as follows:

1. An initial low capillary pressure is selected.
2. All throats which can be invaded at the specified pressure are identified.
3. The selected throats are marked as 'Invaded'.
4. Any throats set as 'Invaded' in previous step, but if it is not connected to the injection face or pathway of invaded pores and throats, are set back to a 'Not invaded'.
5. The algorithm proceeds until the entire domain is saturated.

The OP algorithm scans all clusters that can be invaded based on the initial capillary pressure. The capillary pressure is increased stepwise and all throats and pores connected to the source of invading fluid are invaded at the same time. In terms of computational efficiency, the calculation of this algorithm is performed based on clusters of connected pores, which requires less computational cost. For liquid water intrusion, the same algorithm may be used but the bottom face of the network is assigned as a boundary.

2.2.2.2 Invasion percolation

Water intrusion is also described using the invasion percolation (IP) algorithm presented by Wilkinson and Willemsen [105] and implemented in OpenPNM. The bottom face ($z = 0$) is used as a boundary for water intrusion. The IP algorithm proceeds by scanning throats connected to the boundary face and invading them with the lowest capillary pressure. Appendix B provides codes for the IP simulation. The IP algorithm works as follows:

1. Inlet pores are selected from the boundary face
2. Throats connected to the inlet pores are added to a list and sorted based on entry capillary pressure.
3. A throat with the lowest capillary pressure and the pore connected to it are invaded.
4. The throats connected to the newly invaded pore are added to the list for the next step.
5. Invasion continues until the domain is fully occupied by the invading phase.

As mentioned above, the intrusion algorithm represents drainage of a wetting phase invaded by a non-wetting phase. Resistance is based on the capillary pressure of each throat, which means the invading process starts with an initial dry network and sequentially invades accessible throats of least resistance. In the general case of invasion percolation, when a throat is invaded, the pores connected with the throat are spontaneously filled, since pores are larger than throats. Invasion percolation progresses one element at a time by tracking a list of non-invaded throats connected to the invaded pores. This enables an invasion sequence to be determined, which allows a more precise prediction of liquid water break through at each saturation point.

The major difference of IP from OP is that IP progresses by invading one pore at a time. Figure 2.8 shows difference between OP and IP. Both algorithms show nearly identical saturated portions when Sat.$= 0.17$ and Sat.$= 0.77$. The major difference is shown when Sat.$= 0.46$, which appears to be the first breakthrough saturation. The OP algorithm could not capture this saturation point, since there is no point between Sat.$= 0.17$ to 0.77. For this reason, even though IP requires more computational resources, it is more appropriate to evaluate the breakthrough point of the simulation.

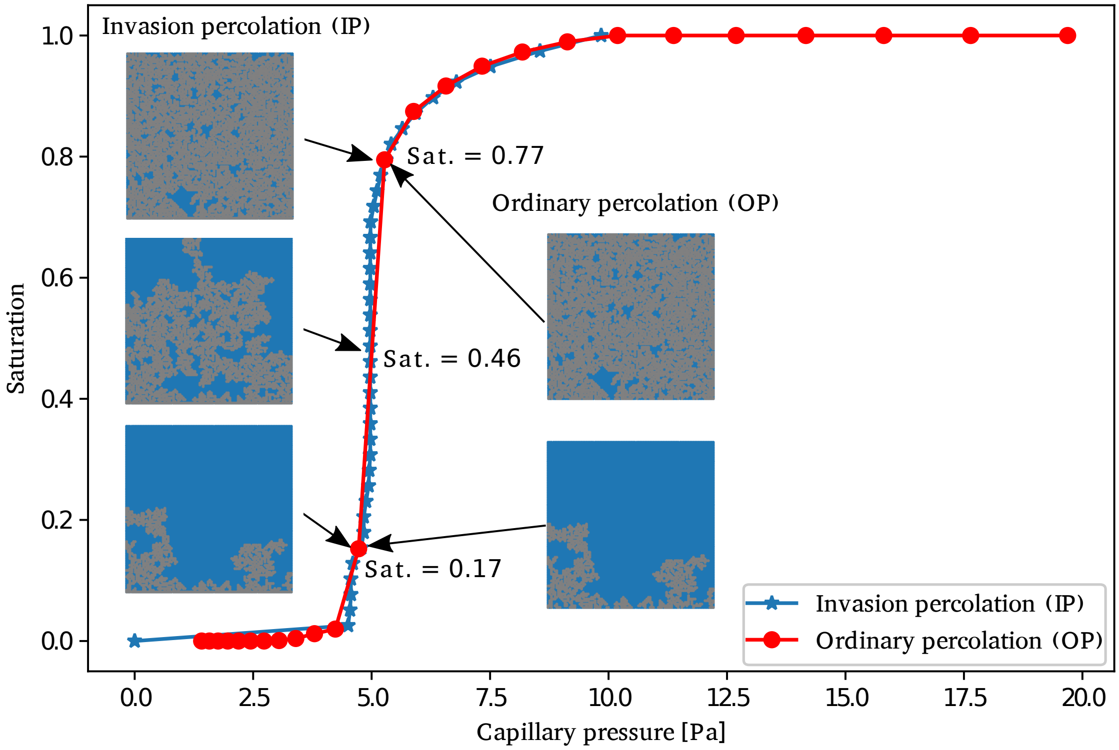


Figure 2.8 – Illustrative comparison of the invasion percolation (IP) and ordinary percolation (OP) algorithms where gray represents the intruding phase. The result from IP is able to capture the first breakthrough point at Sat.= 0.46, whereas that from OP show sudden jump from Sat.= 0.17 to 0.77.

In this work, MIP simulation is done only using the OP algorithm, whereas the liquid water intrusion is simulated using both OP and IP. For both intrusion cases, the process is assumed to be quasi-static and capillary driven.

2.2.3 Diffusion

In order to estimate diffusivity, the Fick's law module in OpenPNM is used. Fick's first law of diffusion states that:

$$N_a = c_t D_a \nabla x_a \tag{2.13}$$

where N_a is flux of species, c_t is the bulk concentration, D_a is the molecular diffusivity, and x_a is the mole fraction of species a. This relation is employed to define the diffusion between pore i and j in a pore network. Equation (2.13) can be discretized as:

$$N_{a,ij} = \frac{n_{a,ij}}{A_{ij}} = \frac{c_t D_a}{l_{ij}}(x_{a,i} - x_{a,j}) \tag{2.14}$$

where $n_{a,ij}$ is the molar flux between pores i and j, and A_{ij} and l_{ij} are the cross-sectional area and length of the pore and throat conduit. The diffusive conductivity, g_{ij} between two neighbouring pores i and j through the connecting throat is defined as:

$$g_{ij} = \frac{cDA_{ij}}{l_{ij}} \tag{2.15}$$

The conductivity is used in Fick's law to define the diffusive transport between pore i and j:

$$n_{a,ij} = g_{ij}(x_i - x_j) \tag{2.16}$$

In order to estimate diffusive mass transport in the entire network, a resistance network is constructed where the following material balance is performed at every pore, i,

$$\sum_{j=1}^{n} g_{ij}(x_i - x_j) = 0 \tag{2.17}$$

where j are all the neighbouring pores, g_{ij} is the conductance between pores and throat conduit, and x is the molar fraction at each pore. The net diffusive conductance, g_{ij}, of pore and throat assembly is calculated as:

$$g_{ij} = \left(\frac{1}{g_{p,i}} + \frac{1}{g_{t,ij}} + \frac{1}{g_{p,j}}\right)^{-1} \tag{2.18}$$

where $g_{p,i}$, $g_{p,j}$ and $g_{t,ij}$ represent the conductance of each pore i, pore j and throat ij, which are defined as:

$$\begin{aligned} g_{p,i} &= \frac{cD_g A_{p,i}}{r_{p,i}} \\ g_{t,ij} &= \frac{cD_g A_{t,ij}}{l_{t,ij}} \end{aligned} \tag{2.19}$$

where $l_{t,ij}$ is the length of the throat, $r_{p,i}$ is the radius of the pore, $A_{p,i}$ and $A_{t,ij}$ are the cross sectional area of pore i and throat ij, c is the bulk concentration, and D_g is the bulk diffusivity of the gas.

The effective diffusion coefficient can be calculated as from the predicted net flow rate as:

$$D_a^{eff} = \frac{N_a^{out} L}{cA(x_a^{in} - x_a^{out})} \tag{2.20}$$

where N_a^{out} is the flow rate of species a at the outlet surface, L is the length of the porous media, and A is the total cross sectional area of the entire porous media. Appendix A provides sample code to perform diffusion simulations.

2.2.4 Permeation

To estimate permeability, the OpenPNM fluid flow module is used. The convective transport in each pore can be approximated by Hagen-Poiseuille equation in a cylindrical tube:

$$Q = \frac{\pi r^4}{8\mu l}\Delta p \tag{2.21}$$

where Δp is the pressure difference between the two ends, l is the length of pipe, μ is the dynamic viscosity, Q is the volumetric flow rate, and r is the pipe radius. Employing this relation to pores i and j, equation (2.21) can be written as:

$$q = \frac{\pi r_{ij}^4 (p_i - p_j)}{8\mu l_{ij}} \tag{2.22}$$

where p_i and p_j are the pressures in pores i and j, and l_{ij} and r_{ij} are the length and radius of the throat connecting pores i and j. For simplicity, the pressure loss in each half-pore is neglected [36]. The flow rate given by equation (2.22) can be generalized as:

$$q = g_{ij}(p_i - p_j) \tag{2.23}$$

where g_{ij} is the conduit's hydraulic conductance. Equation (2.23) may be used to construct a resistance network where the material balance at each pore i can be defined as:

$$\sum_{j=1}^{n} g_{ij}(p_j - p_i) = 0 \tag{2.24}$$

where p_i and p_j are the pressures of each pore. The hydraulic conductance, g_{ij}, of the pores and throat assembly can be determined using equation (2.18). The conductance of each pore and throat is defined as:

$$\begin{aligned} g_{p,i} &= \frac{\pi r_{p,i}^4}{8\mu l_{p,i}} \\ g_{t,ij} &= \frac{\pi r_{t,ij}^4}{8\mu l_{t,ij}} \end{aligned} \tag{2.25}$$

where $r_{p,i}$ and $r_{t,ij}$ are the radius of pore i and throat ij.

Once the net flow rate, Q, is obtained by summing the flow rate of outlet pores, the permeability of the network can be found from Darcy's law:

$$K = \frac{Q\mu L}{A(P_{in} - P_{out})} \tag{2.26}$$

where K is the absolute permeability, and P_{in} and P_{out} are inlet and outlet boundary pressures.

$$k = \frac{\dot{m}\mu l}{\rho A \Delta p} \tag{2.27}$$

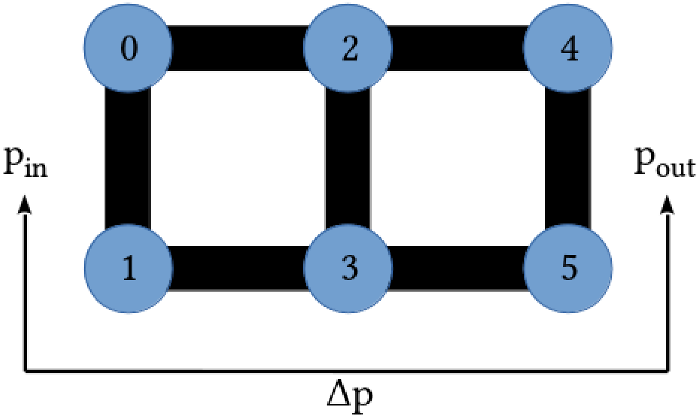


Figure 2.9 – Example 3 x 2 network for building coefficient matrix. Each pore is numbered from 0 to 5.

2.2.5 Solution methodology

Diffusion and permeation can be calculated using the same resistance network based conservation rule, i.e., equations (2.17) and (2.24). In this section, an analytical example is shown to illustrate how the resistance network result in a linear system of equation that can be solved using a linear algebra solvers. To illustrate this case, a simple 3×2 pore network, see Figure 2.9, is employed to describe the methodology solving pressure driven convective flow in the network with inlet and outlet pressure boundary conditions.

The 3×2 pore network, as shown in Figure 2.9, contains 6 pores and 7 throats. The pores on left and right sides have known pressures as a boundary condition.

To analyze the flow between pores, the relation shown in equation (2.24), which is analogous to Kirchhoff's junction rule, is utilized. In Kirchhoff's current law, the algebraic sum of all the currents entering and leaving a junction must be equal to zero. Applying the equivalent material balance rule to the pore network in Figure 2.9 yields:

$$\begin{aligned}
&\text{Pore 0:} \quad g_{01}(p_0 - p_1) + g_{02}(p_0 - p_2) = 0 \\
&\text{Pore 1:} \quad g_{10}(p_1 - p_0) + g_{13}(p_1 - p_3) = 0 \\
&\text{Pore 2:} \quad g_{20}(p_2 - p_0) + g_{23}(p_2 - p_3) + g_{24}(p_2 - p_4) = 0 \\
&\text{Pore 3:} \quad g_{31}(p_3 - p_1) + g_{32}(p_3 - p_2) + g_{35}(p_3 - p_5) = 0 \\
&\text{Pore 4:} \quad g_{42}(p_4 - p_2) + g_{45}(p_4 - p_5) = 0 \\
&\text{Pore 5:} \quad g_{54}(p_5 - p_4) + g_{53}(p_5 - p_3) = 0
\end{aligned} \tag{2.28}$$

By rearranging equation (2.28) with respect to the pressure, the following equations are obtained:

$$\begin{aligned}
&\text{Pore 0:} && p_0(g_{01}+g_{02})+p_1(-g_{01})+p_2(-g_{02})=0\\
&\text{Pore 1:} && p_0(-g_{10})+p_1(g_{10}+g_{13})+p_3(-g_{13})=0\\
&\text{Pore 2:} && p_0(-g_{20})+p_2(g_{20}+g_{23}+g_{24})+p_3(-g_{23})+p_4(-g_{24})=0\\
&\text{Pore 3:} && p_1(-g_{31})+p_2(-g_{32})+p_3(g_{31}+g_{32}+g_{35})+p_5(-g_{35})=0\\
&\text{Pore 4:} && p_2(-g_{42})+p_4(g_{42}+g_{45})+p_5(-g_{45})=0\\
&\text{Pore 5:} && p_3(-g_{53})+p_4(-g_{54})+p_5(g_{54}+g_{53})=0
\end{aligned} \tag{2.29}$$

In matrix form, the equations above can be written as:

$$\begin{bmatrix} A & -g_{01} & -g_{02} & 0 & 0 & 0 \\ -g_{10} & B & 0 & -g_{13} & 0 & 0 \\ -g_{20} & 0 & C & -g_{23} & -g_{24} & 0 \\ 0 & -g_{31} & -g_{32} & D & 0 & -g_{35} \\ 0 & 0 & -g_{42} & 0 & E & -g_{45} \\ 0 & 0 & 0 & -g_{53} & -g_{54} & F \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.30}$$

where

$$\begin{aligned}
A &= g_{01}+g_{02},\\
B &= g_{10}+g_{13},\\
C &= g_{20}+g_{23}+g_{24},\\
D &= g_{31}+g_{32}+g_{35},\\
E &= g_{42}+g_{45},\\
F &= g_{54}+g_{53}
\end{aligned}$$

Next, the boundary conditions must be applied. The pressure boundary conditions given in Figure 2.9 are:

$$\begin{aligned}
p_0 &= p_1 = p_{in}\\
p_4 &= p_5 = p_{out}
\end{aligned}$$

Therefore, the system of equation reduces to:

$$\begin{bmatrix} C & -g_{23}, \\ -g_{32} & D \end{bmatrix} \begin{bmatrix} p_2, \\ p_3 \end{bmatrix} = \begin{bmatrix} g_{20}p_{in} + g_{24}p_{out}, \\ g_{31}p_{in} + g_{35}p_{out} \end{bmatrix} \tag{2.31}$$

The reduced system is solved by OpenPNM using the scipy direct solvers within python. This methodology was used to solve steady state transport with no reaction.

2.2.6 Multiphase transport

Multiphase transport is simulated by solving the governing equations in each phase separately. To do this, the conductance values, i.e., hydraulic or diffusive conductance, of pores and throats occupied by the liquid phase are severely reduced. This is done by multiplying a factor of 10^{-6} the original conductance values. In this way, the gas phase cannot pass through the pores and throats when saturated by the liquid phase. The gas transport will occur through the non-saturated pores only.

Likewise, the liquid phase transport properties can be determined based on the invasion steps made by the percolation algorithm. When computing transport properties for the liquid phase at a specific saturation level, the conductance values of saturated pores and throats are used for transport simulations, and non-saturated regions are restricted.

In this way relative transport characteristics for the gas and liquid phases can be determined by running transport simulations successively, while the percolation algorithm is running.

2.3 Direct numerical simulation

This section describes the continuum based numerical models in OpenFCST [96] used to simulate transport in porous media. In DNS, the transport equations are solved directly on a computational mesh generated based on the microstructure images. Before running simulations an algorithm is used to enhance the mesh by adding pore size data in each cell. This data will also be used to perform the intrusion simulation based on the capillary pressure of each pore. To calculate pore data and virtual intrusion in the porous media, two algorithms, proposed by Sabharwal et al. [29, 30] and implemented in OpenFCST, are used, i.e., a sphere fitting algorithm and a full morphology (FM) algorithm.

2.3.1 Pore size distribution

Computing a pore size distribution (PSD) of the porous media is the first step of the pore scale modelling since the pore data is required to obtain the enhanced computational mesh and to run intrusion simulations using the FM algorithm. Pore size data can be obtained using a sphere fitting algorithm. Computing the PSD of a given microstructure requires the Euclidean distance transform at each pore voxel. The Euclidean distance transform finds the nearest point of the solid phase for each pore voxel and stores the distance between the pore and nearest solid voxel. The Euclidean

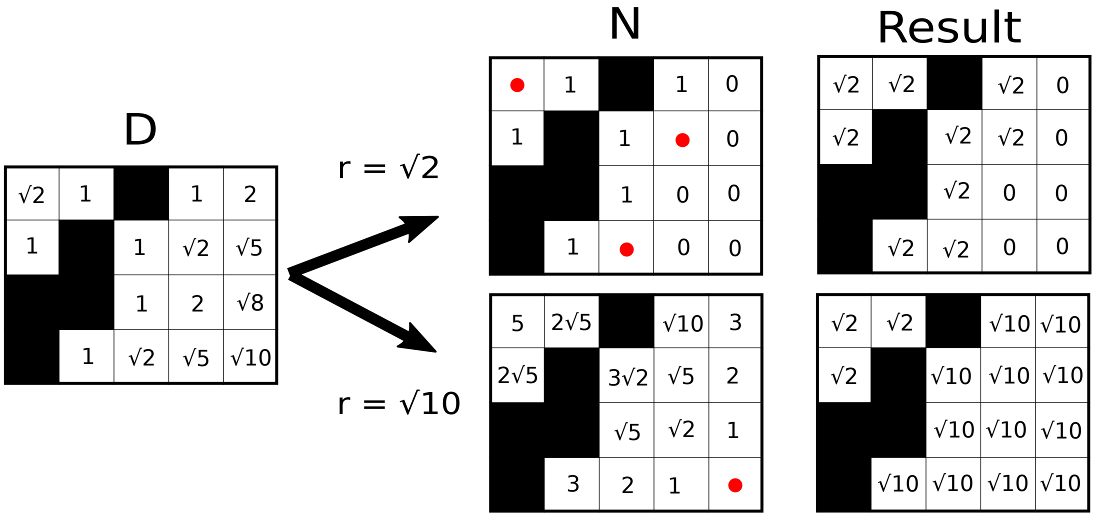


Figure 2.10 – Illustrative explanation for computing the pore size distribution.

distance value at each pore voxel represents the maximum sphere radius, which can be inscribed within the pore region as the centre of the sphere. The algorithm computes pore radius and feasible locations by the following steps:

1. A Euclidean distance transform (D) is computed using the scipy $distance-\_transform\_edt$ function.

2. A distance value in the distance map is selected as a radius value.

3. The algorithm then finds all locations where $D >$ radius.

4. Another Euclidean distance transform map (N) is computed on the neighbourhood to the selected locations in previous step.

5. Find locations in the distance map N with radius smaller than that in distance map D and add these locations in pore distribution matrix P with the radius value.

6. Repeat 2–5 until the radius reaches the maximum value in the distance transform matrix (D).

Figure 2.10 shows how the sphere fitting algorithm assigns a radius value to each pixel. Based on the Euclidean distance map, D, the algorithm tries to find the feasible locations where voxels with the selected radius (r) can fit. Once the feasible locations are identified, the value of radius are assigned to the voxels within the N.

For example, when r is $\sqrt{2}$, values in D larger than $\sqrt{2}$ become 0, and the algorithm computes another distance map, N, based on 0 points. In the distance map, N, all values smaller than $\sqrt{2}$ are converted to $\sqrt{2}$, and the converted values are used for radius of each pore voxel. When r is $\sqrt{10}$, the same procedures generate the N distance map in the bottom line of Figure 2.10. As the three values at the left-top side of N are larger than $\sqrt{10}$, they are not converted to $\sqrt{10}$, and remain at $\sqrt{2}$ as assigned in the previous step. For this reason, every pore voxel has a pore size information which can be either the centre of a sphere or part of a larger sphere.

2.3.2 Meshing

Unlike the PNM approach which approximates pore space to obtain the pore network, in direct numerical simulations, the binary image is directly converted to a 3D unstructured hexahedral VTK mesh. The VTK mesh is generated using a python based module in OpenFCST developed by Sabharwal et al. [29] that converts each voxel in the image to a mesh element, thereby a hexahedral geometry mesh is generated using the TVTK python library. The VTK conversion can be done with multiple material IDs at the same resolution and additional information can be passed to every cell in the form of “field data”. In this way the computed pore sizes can be saved in each cell of the mesh.

A stack of binarized μCT images is used to generate a 3D mesh which can then be used in OpenFCST for transport simulations. To reduce the unwanted isolated region in the void area, the percolating void phase is first identified. The cluster identification proposed by Hoshen and Kopelman [106] and the percolating network extraction algorithm proposed by Pant [100] are used. First, a cluster label is assigned to each pixel of the target phase. Isolated regions are found at this stage and individually labeled. Then, all labels assigned to the isolated regions are replaced by equivalent minimum labels, which is the label of solid region in this case. The extracted percolating clusters are used for mesh generation. Appendix D provides sample code for the mesh generation.

2.3.2.1 Image coarsening

Given the size of the original image stacks, which are 600 × 600 × thickness, δ_t voxels, it is computationally expensive to run Stokes flow simulations. This is because velocities in all three dimensions, as well as the pressure, must be resolved. To circumvent this issue, the original domain is cropped and coarsened [100]. Each subsample is firstly cropped to 400 × 400 × δ_t voxels and the cropped stack is then

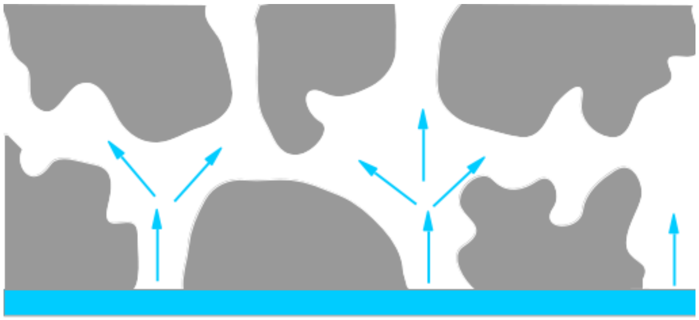


Figure 2.11 – Illustration of the boundary based water injection.

coarsened to 200 × 200 × δ_t voxels. The in-plane length, 400 voxels, is selected to adequately account for the in-plane pore sizes while coarsening the stack is done to reduce the computational cost required for the simulation.

Image coarsening can be performed using the Porous Media Stochastic Reconstruction Tool (PMSRT) using a nearest neighbor interpolation method. The algorithm for resizing a binary image developed and implemented by Pant is discussed in reference [100]. Appendix E provides sample code for PMSRT running.

2.3.3 Intrusion

The cluster based full morphology (FM) method proposed and implemented in OpenFCST by Sabharwal is used for intrusion simulations [30]. The algorithm is used for both water and mercury intrusion. The algorithm provides three different intrusion modes. The first two modes are for water intrusion termed boundary mode and nucleation mode. The boundary mode injects water from the bottom boundary face while the nucleation mode intrudes water within the domain. The third and last mode is mercury intrusion which injects mercury from all external 6 faces.

Figure 2.11 represents an illustration of water intrusion using the boundary mode. The bottom face is selected as a boundary face and water is intruded if the liquid pressure at the boundary is higher than the critical intrusion pressure for the pore. The critical intrusion pressure is computed using the Washburn equation given by equation (2.12). The boundary mode is used for water intrusion into PTLs and MIP for CLs.

The nucleation mode is used to introduce water within the CL porous media. In the CL of PEMFC, water molecules are produced as the product of the oxygen reduction reaction within the layer. Therefore, the nucleation mode may be more appropriate to represent the water intrusion in the CL.

Running intrusion simulations requires several inputs to specify the characteristics of the material and the liquid. The input parameters are the contact angle, the intrusion mode, and the number of steps for computing the pore size distribution. Based on the inputs, the FM algorithm proceeds with the following steps:

1. Calculate the pore size distribution of the sample based on the algorithm described in Section 2.3.1.
2. The critical intrusion pressure of each pore is calculated based on the pore radius obtained from PSD and the contact angle.
3. Based on the capillary pressure of each pore, all pore clusters with lower critical intrusion pressure than the liquid pressure are scanned.
4. If the selected pores are connected to the existing liquid filled pores, the clusters are set to 1, which means intruded, and stored as an array of liquid filled (intruded) pores.

These steps are continued until all pores in the microstructure are intruded. More details about the algorithm can be found in reference [30]. Figure 2.12 shows various stages of liquid water intrusion in a PTL image sample using the boundary mode. It can be seen that the bottom face is selected as a boundary face and liquid water is intruded based on the capillary pressure of the bottom pore spaces.

2.3.4 Diffusion

Mass transport is governed by Fick's second law. In the absence of a chemical reaction, the mass conservation for oxygen is:

$$\nabla \cdot (D_a c_t \nabla x_a) = 0 \tag{2.32}$$

where D_a is the diffusion coefficient, c_{tot} is the total gas concentration, and x_{O_2} is the molar fraction of species a. The boundary conditions are specified as follows:

$$\begin{aligned} x_a &= x_a^{in} \text{ on } \mathit{\Gamma}_1, \\ x_a &= x_a^{out} \text{ on } \mathit{\Gamma}_2, \\ (D_a c_t \nabla x_a) \cdot n &= 0 \text{ everywhere else}, \end{aligned} \tag{2.33}$$

where $\mathit{\Gamma}_1$ is the inlet plane and $\mathit{\Gamma}_2$ is the outlet plane opposite to the inlet plane, and x_a^{in} and x_a^{out} are Dirichlet boundary conditions at the inlet and outlet faces.

The governing equation is solved using a finite element solver developed in our laboratory, open source fuel cell simulation toolbox, OpenFCST [96]. In order to

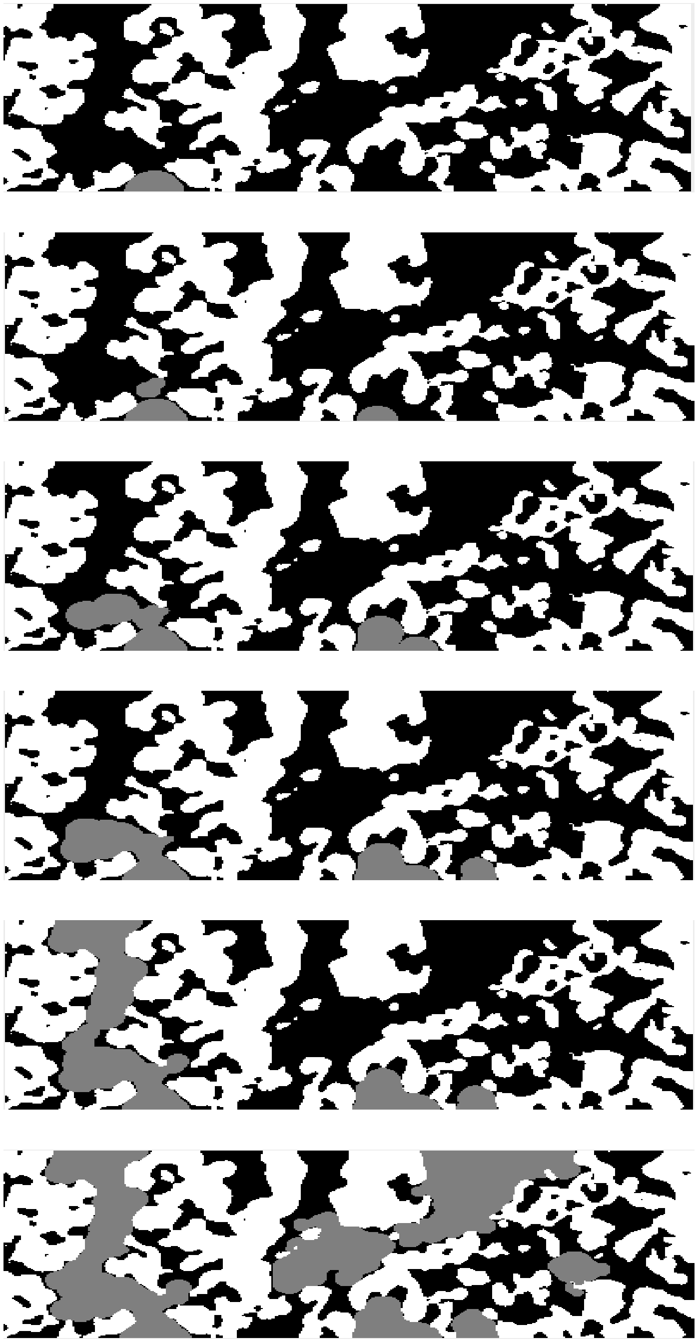


Figure 2.12 – Sections of a sample PTL image at various stages of intrusion algorithm. Black represents pore space, white solid, and grey liquid water.

solve the partial differential equation, the equation is first discretized using the finite element method, the boundary conditions are applied, and the linear system of algebraic equations is solved using an iterative solver, Conjugate gradient (CG). Once the solution has been obtained, the flux of the species is calculated at post-processing at the boundary of the domain using Fick's first law given in equation (2.13). By integrating the flux over the outlet surface, the net flow rate of species can be obtained. The effective diffusion coefficient can then be calculated using equation (2.20). In this work, the simulations were performed at a constant temperature 353 K and a constant pressure of 1 atm.

2.3.4.1 Solution methodology

The governing equation in Section 2.3.4 is solved with the boundary conditions given in equations (2.33). The governing equation is discretized using the Bubnov-Galerkin method. The governing equation is multiplied by scalar test function, v, and integrated over the domain, Ω, to obtain the weak form.

$$\int_{\Omega} v[\nabla \cdot (A\nabla u)]d\Omega = 0 \tag{2.34}$$

where A is a transport coefficient, i.e. air diffusivity, and u is the solution function. The LHS of equation (2.34) can be rearranged using tensor algebra as:

$$-\int_{\Omega} v[\nabla \cdot (A\nabla u)]d\Omega + \int_{\Omega} \nabla \cdot (vA\nabla u)d\Omega = 0 \tag{2.35}$$

By applying the divergence theorem, the LHS can be simplified to:

$$-\int_{\Omega} v[\nabla \cdot (A\nabla u)]d\Omega + \int_{\Gamma} (vA\nabla u) \cdot \mathbf{n}d\Gamma = 0 \tag{2.36}$$

where Γ represents the boundaries and $\mathbf{n}$ is the normal direction to the boundary. Next, a finite element approximation is used such that v and u are approximated by

$$u = \sum \phi_i u_i \quad \text{and} \quad v = \sum \phi_j v_j \tag{2.37}$$

where $\phi_i(x, y, z)$ are first order Lagrange elements. After substituting the approximation functions, a linear system of algebraic equation is obtained for u_i. The linear system is solved using the Conjugate Gradient (CG) solver. The module for solving diffusion in a 3D microstructure is developed and implemented by Sabharawal et al. [29] in OpenFCST.

2.3.5 Permeation

Fluid flow of very low Reynolds number characteristic of transport in a PTL is governed by the incompressible Stokes flow equation:

$$\begin{aligned} -\mu\nabla\cdot[(\nabla\mathbf{u})+(\nabla\mathbf{u})^T]+\nabla p=f \\ \nabla\cdot\mathbf{u}=0 \end{aligned} \tag{2.38}$$

where $\mathbf{u}$ denotes the velocity of a fluid, p is the pressure, f are external forces, and μ is the dynamic viscosity. The governing equation is subject to the following Dirichlet boundary conditions:

$$\begin{aligned} &p=p_{in} \text{ on } \mathit{\Gamma}_1 \\ &p=0 \text{ on } \mathit{\Gamma}_2 \\ &\mathbf{u}=0 \text{ on } \mathit{\Gamma}_2 \end{aligned} \tag{2.39}$$

where p_{in} is the assigned pressure on the inlet face. The governing equation is discretized with the finite element method and the system is solved using the Conjugate Gradient (CG) solver in OpenFCST. The effective permeability can be calculated using equation (2.26) with the net flow rate at the outlet face.

Chapter 3

Results and Discussion

This chapter describes the analysis of obtained μCT images and transport studies carried out using the numerical tools described in Chapter 2. The objective of this study is to understand the difference between the two different approaches for analyzing 3D microstructure characteristics, and to estimate transport properties, using μCT images. Section 3.1 presents a discussion of image acquisition results. This section explains image binarization results and the assessment of segmentation quality, as well as microstructure characterization using statistical analysis and intrusion algorithms. Section 3.4 and Section 3.5 estimate dry transport properties of three PTL samples and a CL stochastic reconstruction using PNM and DNS methods. The predicted results are compared to experimentally measured results. Section 3.6 presents relative transport properties of partially wetted samples.

3.1 Image analysis

Three PTL samples, i.e., Toray 120C, SGL 39BA, and sintered Ti, were imaged using μCT. These materials are selected due to their microstructural differences. Fibrous carbon papers are widely used for PEMFC GDLs and PEWE cathode PTLs. Toray 120C has carbon fibers with 10% PTFE, whereas SGL 39BA has carbon fibers with 5% PTFE, and a porous carbon matrix between the fibers to hold them together. Due to the difference in internal structures, they have different porosity and pore size distribution. Sintered Ti has been studied widely as PTL for the PEWE anode side. Since the sintered Ti is manufactured based on the Ti powder, its microstructure is completely different from the fibrous materials mentioned above, and has a lower porosity than the fibrous materials.

The μCT image reconstruction of dry and wet Toray 120C GDL with 10% PTFE content were previously published in ref. [30, 107]. The same Toray 120C dry image reconstruction was used in this study. Other samples were scanned at the UofA and reconstructed using the method described in Chapter 2. The originally prepared reconstruction sample size was 1500 × 1500 × thickness, δ_t. The raw images obtained from μCT were segmented using either the Otsu or Sauvola algorithms. The binarized original stacks were then cropped into three subsamples. Each of the three subsamples has dimensions of 600 × 600 × δ_t.

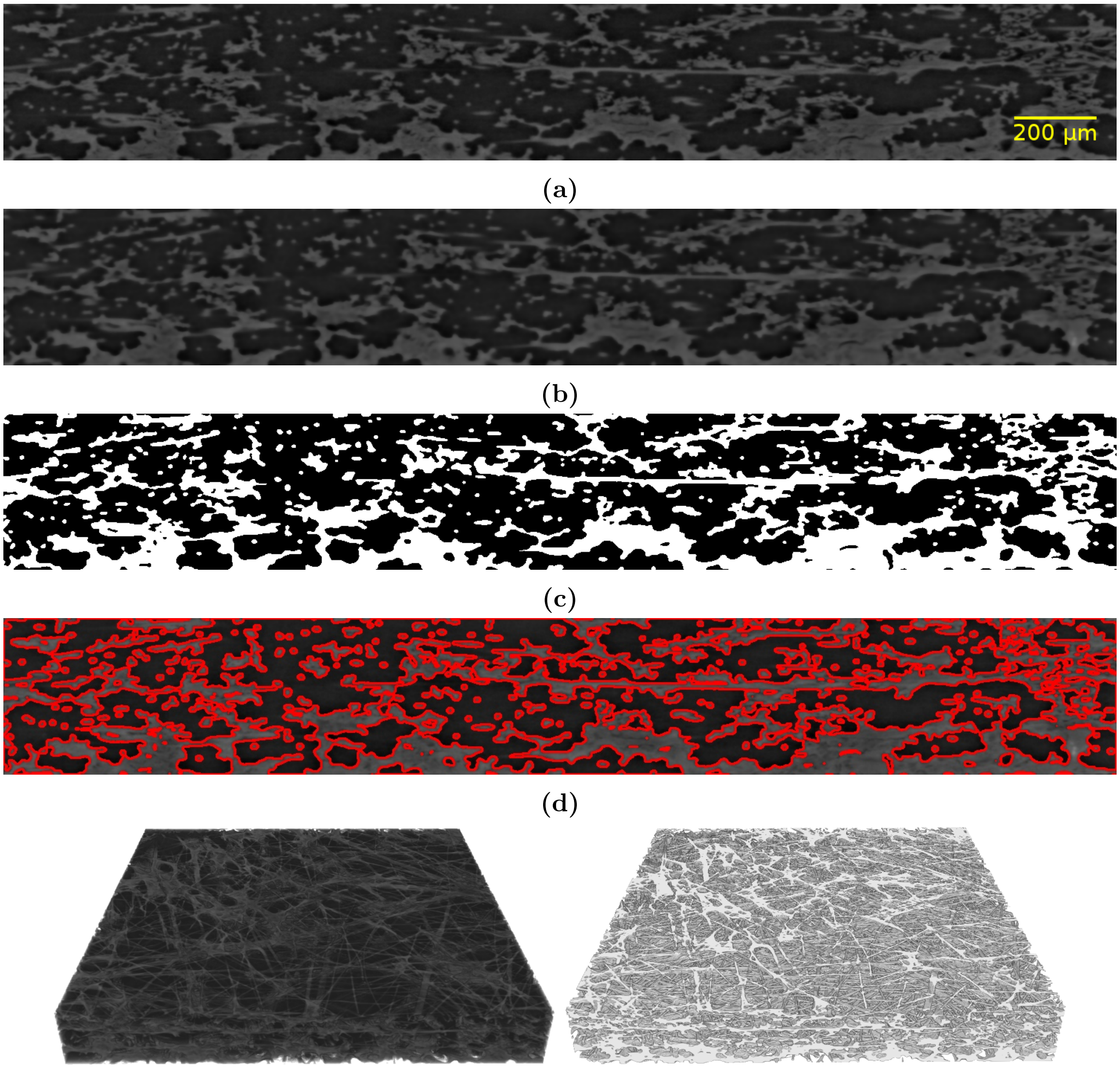
In this section, microstructural characteristics of three subsamples were analyzed using statistical functions. Pore size distributions (PSDs) obtained from PNM and FM were compared. The cumulative volume fraction of each sample was predicted using mercury intrusion porosimetry (MIP) algorithms from both PNM and FM methods. Other PSDs were obtained by taking the derivative of the mercury intrusion curves (MIP-PSD). Finally, PSDs and MIP-PSDs were compared to experimental results.

3.1.1 Image segmentation

The original samples, with a full domain size of 1500 × 1500 × δ_t voxels, were first segmented. Figure 3.1 shows the reconstructed image of each step of the segmentation process for Toray 120C μCT sample. Figure 3.1a shows a raw sectional image of Toray 120C. The raw image shows clearly divided solid and void phases. Figure 3.1b shows the image after the application of the bilateral filter. Applying the bilateral filter on the raw images significantly decreases the noise in both void and solid regions and makes the grayscale intensity of the solid region more uniform. In this case, the quality of the filtered image in Figure 3.1b is similar to that of the raw image, but the solid part of the filtered image has more regularity than the raw image. Figure 3.1c shows the image after binarization. Otsu algorithm was applied to segment the stack shown in Figure 3.1c. The segmentation quality can be checked by overlaying the edges of the segmented images onto the filtered images. The edge overlaying image is shown in Figure 3.1d. The figure shows that the solid and void phases are segmented correctly. The porosity can be computed by:

$$\varepsilon = \frac{V_{void}}{V_{bulk}} = \frac{N_{void}}{N_{bulk}} \tag{3.1}$$

where V_{void} and V_{bulk} are the volume of void region and the entire domain respectively, and, N_{void} and N_{bulk} are the number of voxels in the void region and the entire domain of the binarized image. The thickness of the stack can be determined by the number of pixels in the z-direction. The segmented Toray 120C stack has a porosity of 66% and a thickness of 280 μm in agreement with the porosity and thickness in reference



(a)

(b)

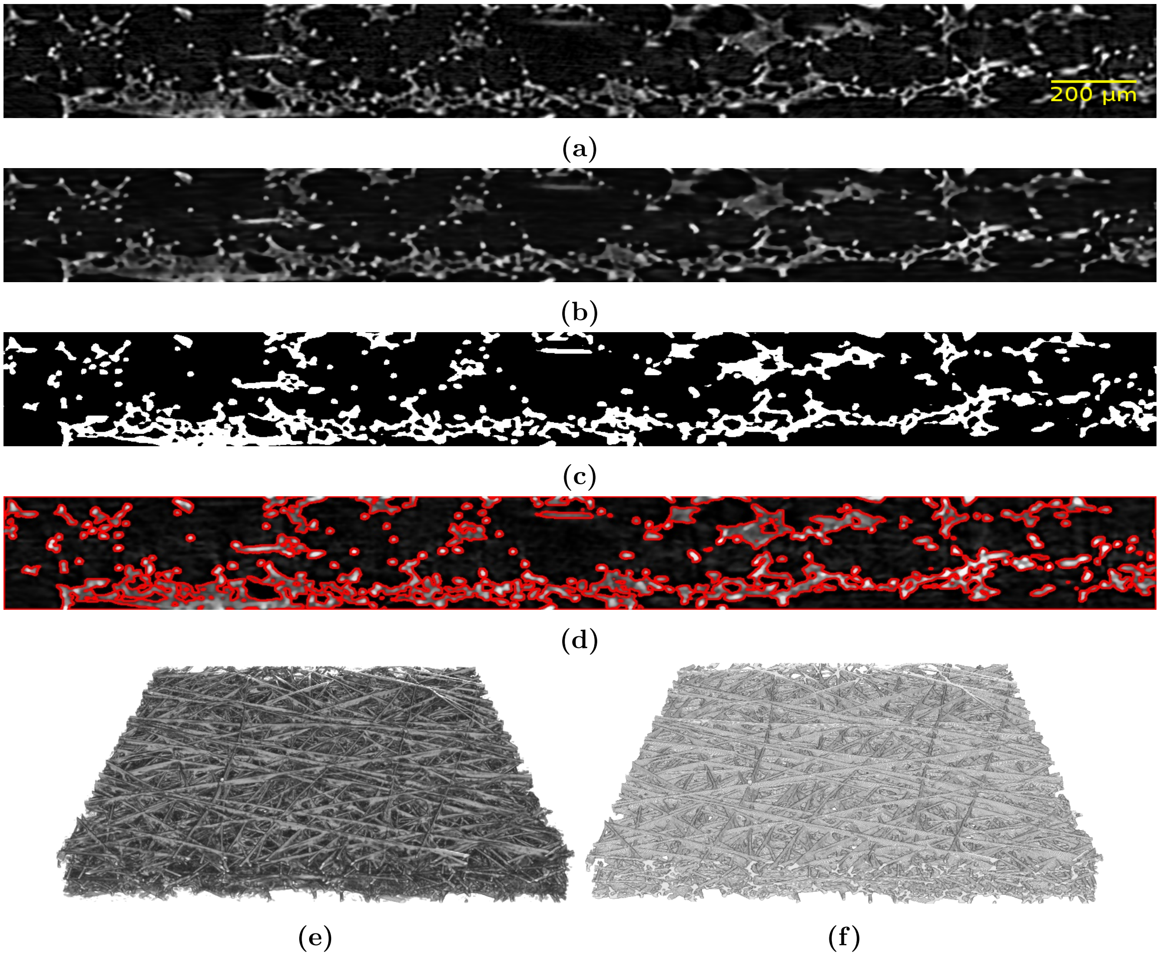
(c)

(d)

(e) **(f)**

Figure 3.1 – Sections of a Toray 120C image at various stages of the image processing: a) raw reconstructed image aligned and cropped, b) after applying 3D bilateral filter, c) segmented using the Otsu's algorithm, d) overlay result, e) raw image in 3D view, and f) segmented image in 3D view.

[30], i.e., 65.9% and 282 μm (Note that the values are not the same as reported by manufacturer due to cropped surface.). The thickness and the porosity of the Toray 120C are slightly smaller than the measured values. This is because the surface region of the original sample was cropped. The core region was used in this study because the same thickness samples were employed in literature [30]. The same thickness and porosity were used in order to be able to compare to previous studies in literature for validation purposes. The pixel resolution of the reconstruction stack was 1.33 μm in



(a) (b) (c) (d) (e) (f)

Figure 3.2 – Sections of a SGL 39BA image at various stages of the image processing, a) raw reconstructed image aligned and cropped, b) after applying 3D bilateral filter, c) segmented using the Otsu's algorithm, d) overlay result, e) raw image in 3D view, and f) segmented image in 3D view.

all directions.

The raw μCT scan for 39BA is shown in Figure 3.2a. As shown in the figure, the raw image contains a substantial amount of noise in the pore region. This should be smoothed as some of this noise has similar gray levels to some of the solid region, which could correspond to the solid part when segmenting the image. The use of the bilateral filter successfully smoothed the noise as shown in Figure 3.2b, even though some bright pixels remain in the pore region after filtering, and their gray levels are lower than the solid pixels. Next, the Otsu algorithm was used to segment the sample. The result is shown in Figure 3.2c. The experimentally measured porosity of the 39BA is higher than the porosity computed from the CT images, i.e., 89 % vs 80 %, respectively. The reason for the discrepancy is that pores smaller than the voxel resolution exist in the carbon matrix. More specifically, the highest resolution of the

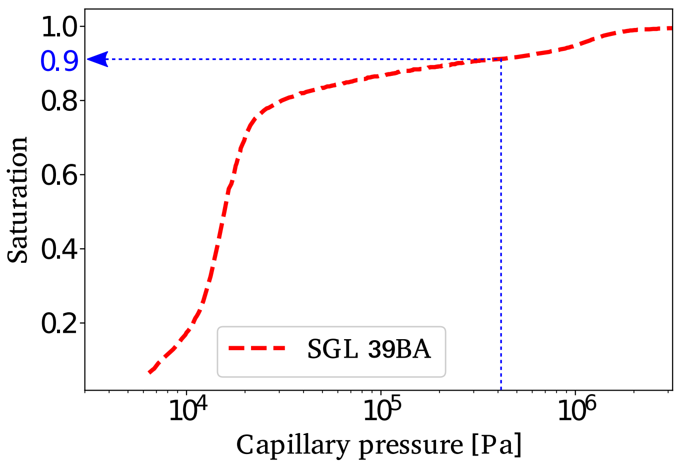
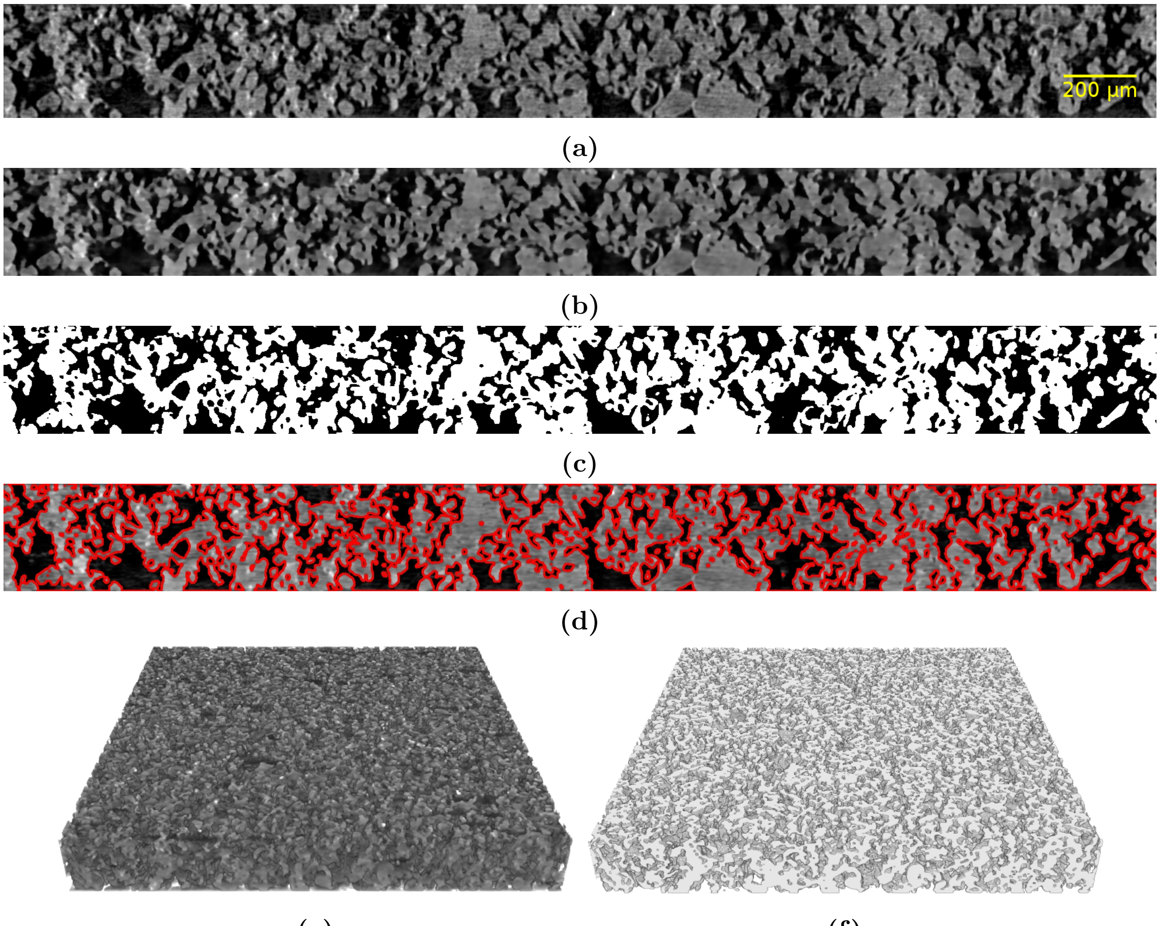


Figure 3.3 – Experimental MIP result for SGL 39BA, showing a pressure, 410 kPa, corresponding to the voxel resolution, 1.79 μm.

Skyscan 1172 is 1 μm, while the pores in the carbon matrix are nanoscale. Hence, fibers, carbon matrix and PTFE are treated as a single solid phase, and the pores smaller than the voxel resolution are neglected. Based on the experimental mercury intrusion data, shown in Figure 3.3, the pressure corresponding to the voxel resolution is around 410 kPa. The volume fraction below the voxel resolution is about 9%, which proves the estimated CT porosity of 80% is in agreement with our observations.

A μCT image of the Ti sample is shown in Figure 3.4a. The image contains more noise in the void and solid regions than the Toray image. The solid region has a rough surface and the void region contains some bright pixels. Those bright pixels might come from the solid information in following slices, called afterimage effect, which should be reduced since the following slices already contain this information. The roughness in one phase could also be noise, which leads to incorrect segmentation result. This noise can be reduced by filtering the images. The bilateral filtered image of the sintered Ti is shown in Figure 3.4b. In the filtered image, the solid phase is smoothed which helps segmentation. As a result, the solid regions have a smoother surface than the raw image and the background noise is also reduced. The Sauvola algorithm with parameters of 20 for window size and 0.5 for the k value was used for segmentation instead of Otsu because the Sauvola algorithm could achieve the desired porosity. The overlaid result in Figure 3.4d shows very good segmentation results.

Given that there is no published literature on the Ti sample used in this study, the thickness was measured from the μCT images, and found to be 292 μm. This value is in agreement with the values of 288 μm obtained from a mercury intrusion



(a)

(b)

(c)

(d)

(e) (f)

Figure 3.4 – Sections of a Ti sinter image at various stages of the image processing, a) raw reconstructed image aligned and cropped, b) after applying 3D bilateral filter, c) segmented using the Sauvola's algorithm, d) overlay result, e) raw image in 3D view, and f) segmented image in 3D view.

Table 3.1 – Comparison of the thickness and porosity of the PTLs.

PTL	CT thickness, δ_t [μm]	Measured δ_t [μm]	CT porosity, ε	Measured ε	Resolution [μm]
Toray 120C	280	362 ± 5.9 [108]	0.66	0.73 [108]	1.33
SGL 39 BA	310	277.8± 7.9 [109], 280 ± 30***	0.80	0.89 ± 0.03*	1.79
Ti sinter	292	288*, 305**	0.46	0.46 ± 0.02*	1.79

* Measured by mercury intrusion porosimetry

** Measured by a micrometer

*** Based on manufacturer's datasheet

experiment and 305 μm measured by a micrometer. The thickness and porosity of each sample are shown in Table 3.1. Results for the samples scanned in our laboratory are in excellent agreement.

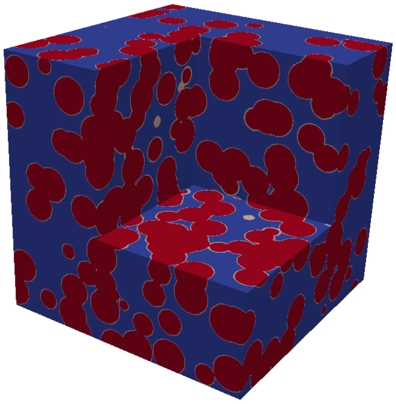


Figure 3.5 – 3D reconstruction of the CL reconstruction image. The pore and solid regions are shown in blue and red, respectively.

3.1.2 Stochastic reconstructions for CL

The overlapping sphere stochastic reconstruction algorithm in OpenFCST was used to generate multiple CL samples. Sabharwal et al. [54] already studied the microstructural characteristics of the CL stochastic reconstructions by comparing the microstructural features with a real CL microstructure obtained from FIBSEM. In order to make a stochastic reconstruction represent a real material, it is important to achieve statistical equivalence of the stochastic reconstructions to a real FIBSEM reconstruction. For this reason, they generated CL stochastic reconstructions with different particle radii and compared statistical correlations with those of the FIBSEM reconstruction. The study concluded that selecting the appropriate particle size had a critical effect on generating statistically equivalent structures and that the stochastic reconstructions with a particle radius 40 nm showed best agreement with the FIBSEM data. As the two-point correlation function and the chord length function in the x and y directions were proven to be nearly identical to the FIBSEM results in the reference [54], in this work, the CL stochastic reconstructions generated with the same conditions mentioned in reference [54] were used.

As discussed in Section 2.1.4, three input parameters were selected for the CL reconstructions: a sphere particle radius of 40 nm, a porosity of 36%, and free overlapping spheres. Figure 3.5 shows the CL stochastic reconstruction result. The domain size was 600 × 600 × 600 nm^3 with a voxel resolution of 2 nm. The stochastic reconstruction samples are used for further studies, such as computing PSD and intrusion simulations.

3.1.3 Statistical analysis

Figure 3.6 shows the two-point correlation functions of subsamples for each PTL case in x (in-plane) and z (through-plane) directions. The two-point correlation functions are equal to the porosity of the sample at a distance of zero and converge to a value of the porosity squared, ε^2, at large distances.

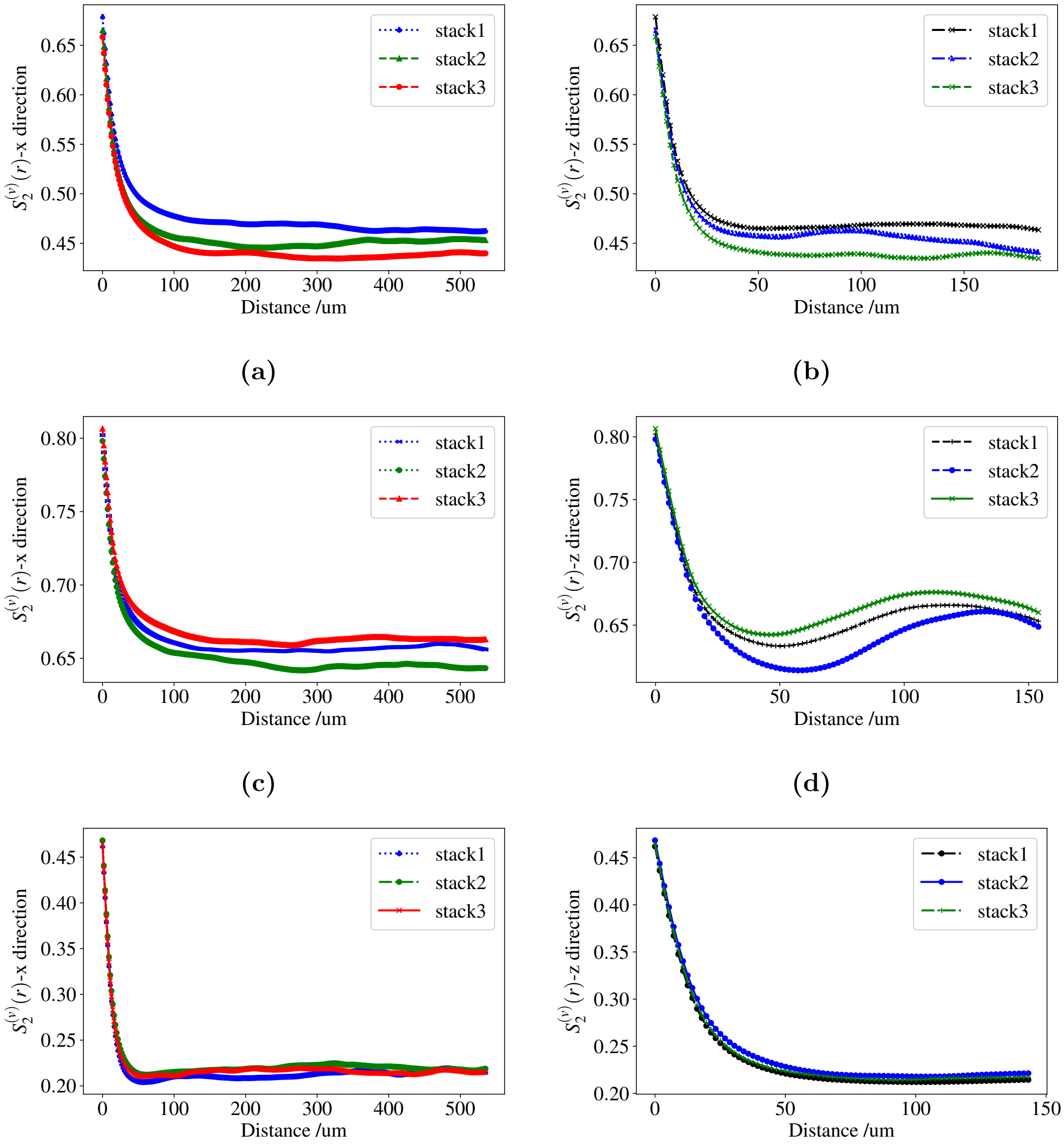
Figure 3.6a and Figure 3.6b show the void phase x and z two-point correlation functions for three Toray 120C subsamples of size 600 × 600 × 210 voxels. The three correlation functions are very similar. The porosity of the original stack is approximately 66% and the individual subsections have porosities that are within 1%.

Figure 3.7 shows the chord length functions of subsamples for each case in x (in-plane) and z (through-plane) directions. As discussed in Section 2.1.5.2, the chord length function contains the length information of the clusters in the image. The statistical functions for the y-direction are not shown here because they have nearly identical distributions as the x-direction.

Figures 3.7a and 3.7b show the chord length functions of Toray 120C subsamples. Each of the chord length functions has identical values, which indicates that each subsample has nearly identical microstructure. Figures 3.7a and 3.7b show that the chord length function is different in the x and z directions. The chord length function in the x-direction spans 0 to 200 μm whereas the chord in the z-direction spans a range of 0 to 100 μm. This highlights the anisotropy of the Toray 120C.

A similar phenomena is seen for the SGL 39BA shown in Figure 3.6c and Figure 3.6d. The fluctuations are significant in the z-direction of the two-point correlation shown in Figure 3.6d. The fluctuation on the graph represents the fact that the binary microstructure is not rigorously statistically homogeneous [110]. The fluctuations would be reduced with a larger domain; however, the trends of the two-point correlation functions are similar for all three subsamples, and it is reasonable to conclude that the subsamples have similar morphology.

For the SGL 39BA results, the chord length function in the x-direction has a non-negligible probability of finding chords of up to 400 μm whereas the probability becomes nearly zero as the chord lengths reach around 300 μm in the z-direction. This also indicates the SGL 39BA sample is anisotropic. The chord length functions of the SGL 39BA subsamples show some probability of finding chords at the longest distance of the graphs, which are about 1000 μm and 300 μm in the in-plane and through-plane directions, respectively. This means there are void chords that span through the size of the image in one direction. This is possible because the top and

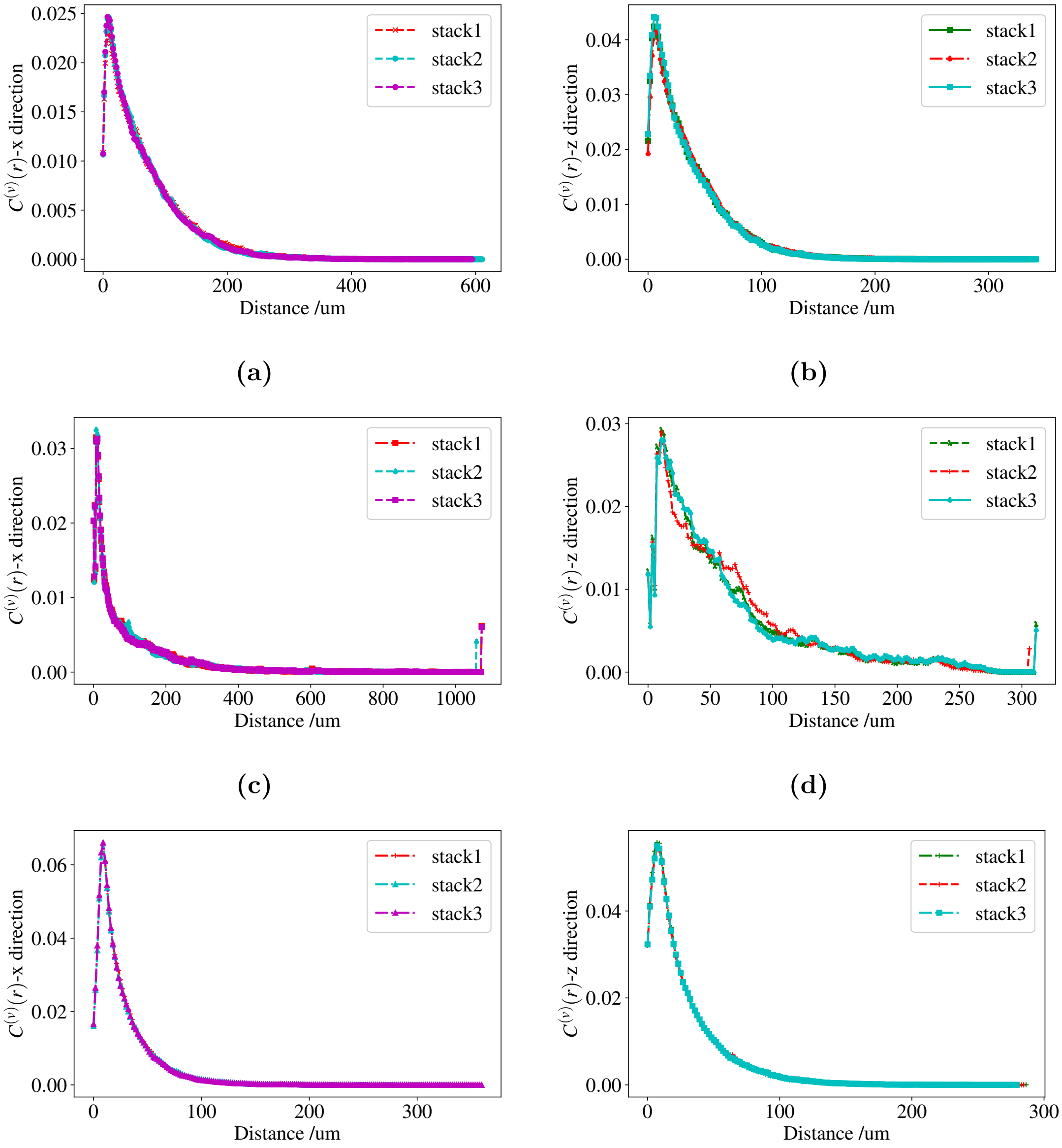


(a) (b)

(c) (d)

(e) (f)

Figure 3.6 – Two-point correlation in the void phase in: a) x and b) z direction for Toray 120C; c) x and d) z direction for SGL 39BA; e) x and f) z direction for Ti.



(a)

(b)

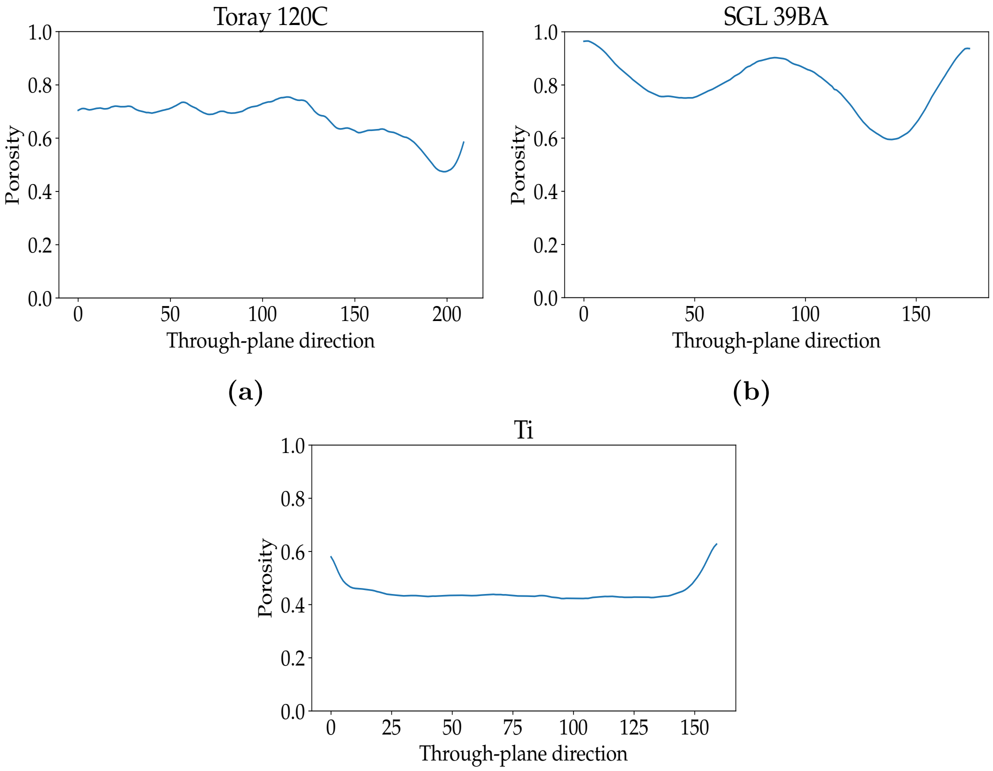
(c)

(d)

(e)

(f)

Figure 3.7 – Chord length function in the void phase in: a) x and b) z direction for Toray 120C; c) x and d) z direction for SGL 39BA; e) x and f) z direction for Ti.



(a) (b) (c)

Figure 3.8 – Porosity profiles for each PTL sample in the z-direction: a) Toray 120C, b) SGL 39BA, and c) Ti.

bottom (surface) regions are not cropped and the original image has high porosity, 80 %. Unlike the porosity profile of Toray 120C shown in Figure 3.8a, the porosity of the 39BA image slices at the top (z=0) and bottom (z=175) regions are higher than 90%, as shown in Figure 3.8b. This explains the images contain mostly void voxels which become the longest chords.

The void phase x- and z-direction two-point correlation functions for the Ti subsamples are nearly identical for each subsamples and in both directions because the Ti material is a powder based PTL, which has a lower porosity and less heterogeneity than the fibrous PTLs.

The chord length functions for the Ti samples do not follow the same trend as the Toray 120C and SGL 39BA as its x- and z-direction chord length functions are very similar to each other. This is due to the homogeneity of the Ti subsamples. The main difference between the x- and z-direction is that the maximum probability value is higher and the probability at values smaller than the maximum is lower in the x-direction than those in the z-direction. However, it is not as significant as in the fibrous PTLs.

Table 3.2 – Average specific interface area and the mean chord length for each subsample.

		Specific interface area ($m^2/m^3 \times 10^{-3}$)				Mean chord length (μm)	
	Subsample	XY	YZ	ZX	Total	(x/y/z)	Porosity (%)
Toray 120C	1	44.06	24.39	25.06	93.51	50.9 / 49.4 / 26.1	67.9
	2	42.96	25.12	24.13	92.20	48.4 / 50.3 / 27.0	66.5
	3	44.44	24.71	25.43	94.58	48.5 / 47.2 / 25.4	65.8
SGL 39BA	1	18.13	13.21	14.38	45.72	107.3 / 98.8 / 64.3	80.2
	2	17.83	12.77	14.05	44.66	103.9 / 99.8 / 63.9	80.6
	3	18.04	13.63	14.25	45.92	111.6 / 101.9 / 65.4	79.8
Ti	1	27.13	31.59	27.55	86.27	26.7 / 30.7 / 27.5	46.2
	2	26.25	30.76	26.73	83.74	27.8 / 32.1 / 28.8	46.8
	3	26.67	31.20	27.03	84.91	27.2 / 31.5 / 28.2	46.4

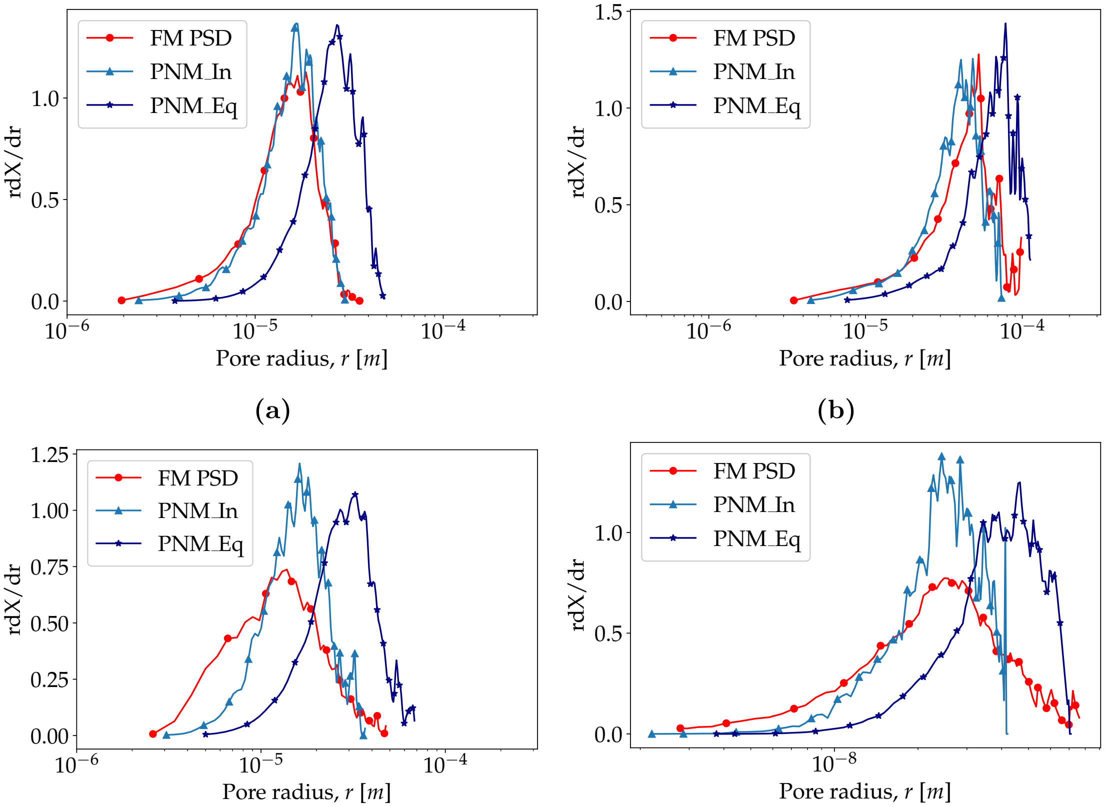
As described in Section 2.1.5.1, the slope of the two point correlation function at $r = 0$ represents the interface area. Table 3.2 shows the specific interface area of each material. The specific interface areas are computed separately for each direction and the total interface area is computed by summing the values in all directions. It is shown that the specific area of each sample has similar values for all material cases.

Table 3.2 shows the mean chord length for each subsample computed by equation (2.10). There are clear differences between mean chord length of the fibrous samples show clear differences in the in-plane and through-plane. This is again an indication of the anisotropy of the fibrous materials. The deviations between subsamples are not significant, which proves the sample size selected is statistically meaningful. As the graphs of Ti showed the least discrepancy between the subsamples, the mean chord length show the smallest discrepancy. This proves higher statistical homogeneity of Ti subsamples than the fibrous materials.

3.1.3.1 Pore size distribution

The pore size distribution (PSD) calculated using the sphere fitting algorithm in OpenFCST and the pore sizes obtained from the network extraction step in PNM are compared in Figure 3.9. As discussed in Section 2.2.1, the PNM network extraction returns two different pore sizes, i.e., inscribed diameter and equivalent diameter. The PSDs from both diameter cases are shown. Since the pore size distribution is a probability distribution function, the area under each curve is 1.

For the PNM results, the equivalent diameter PSD has a larger pore diameter range than the inscribed diameter, since it is calculated based on larger sphere sizes as described in Section 2.2.1. For the fibrous materials, i.e., Toray 120C and SGL 39BA, Figures 3.9a and 3.9b show that the sphere fitting and the inscribed diameter results from PNM have a similar range of pore sizes and pore size probabilities.



(a)

(b)

(c)

(d)

Figure 3.9 – Comparison of pore size distributions for PTL and CL samples obtained from PNM and the sphere fitting algorithm. PNM_In and PNM_Eq represent inscribed diameter and equivalent diameter respectively. a) Toray 120C, b) SGL 39BA, c) Ti, and d) CL

Figures 3.9c and 3.9d show that, on the other hand, large discrepancies between two methods exist for the granular materials, i.e., sintered Ti and CL stochastic reconstructions. For these cases, the sphere fitting method captured more small and large pores than the PNM results. During the network extraction steps, the erroneous peaks which lie on saddles or plateaus of the distance map are removed. Some peaks that are close to the neighbouring peaks may be removed at this step. This may truncate some smaller pores in PNM.

As described in Section 2.3.1, the PSD results from the sphere fitting algorithm are obtained by directly finding locations that fit to a specific radius value on the distance map. As such, the algorithm assigns pore radius data to all voxels in the domain. For this reason, there should be small pores between two relatively big pores to fill the entire domain. This leads to the sphere fitting algorithm returning some portion of voxels with a radius of one voxel length.

One missing piece of information in PSDs from PNM is throat size data. The data obtained from the network extraction in PNM contain pore and throat sizes and the volumes of the pores. If one pore region is made as a result of the watershed segmentation, the volume of the segmented pore region is assigned only for a pore volume. The throats connecting to the pore have size data, but the volume of each throat is zero. As PSDs shown in Figure 3.9 are computed based on differential volume fractions, results from PNM are not able to incorporate throat sizes.

Figure 3.10 shows the pore sizes assigned to the two CL reconstruction slices using the sphere fitting algorithm in OpenFCST, and the pore region segmentation results from the network extraction in Porespy. The first slice was obtained from the y-direction surface of the domain, i.e., Y is equal to 0, and the second slice was from the internal domain, i.e., Y is equal to 150 voxels. The sphere fitting shows small pore regions around the solid (white in the figure), but the network extraction does not show this feature. Each coloured region in the sphere fitting represents a pore size, and different colours in the network extraction indicate each segmented pore region. The segmented pore regions are used to determine pore sizes in the pore network. The inscribed pore diameters are computed by fitting spheres inside the pore regions, whereas the equivalent pore diameters are obtained based on spheres having the same volumes as the pore regions. Three points in the first image of Figure 3.10 are selected to compare the pore sizes between PNM and FM at the surface of the domain. The assigned pore radii at the selected points A, B, and C are 43 nm, 63.6 nm, and 14.3 nm in the sphere fitting case, respectively. The pore diameters at the same points in the network extraction case are 48.8 nm for A, and 56.5 nm for B and C as they are in the same region. The values are used for radii in the sphere fitting, but they are

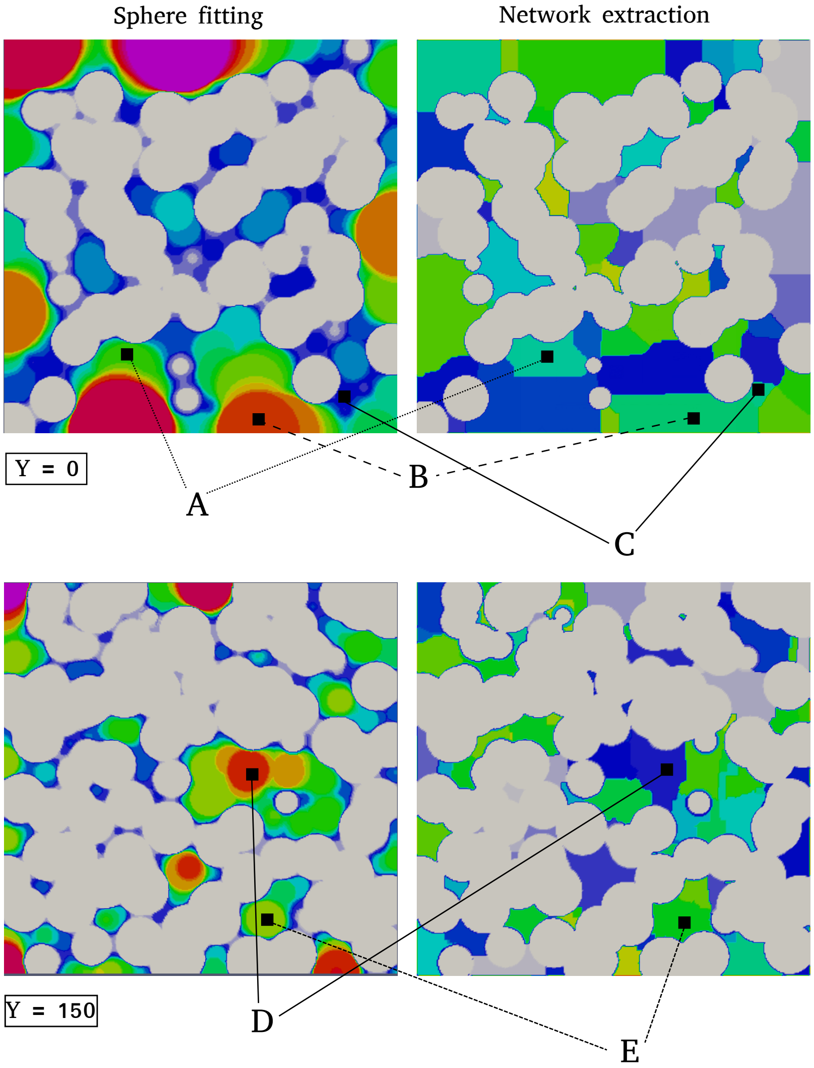


Figure 3.10 – Methods to determine pore sizes in the sphere fitting algorithm and the network extraction. The sphere fitting algorithm assigns a radius value to each voxel. The network extraction segments the entire pore region to separated pore spaces using a watershed algorithm. The different colours represent different radii in the sphere fitting case. The colours in the network extraction represent different pore regions. The pore sizes in PNM are determined by computing a radius that can be inscribed in a segmented pore region. White represents the solid region. Pore radii in the sphere fitting case at the selected points A, B, C, D, and E are 80 nm, 43 nm, 63.6 nm, 34.9 nm, and 26.6 nm. Assigned pore diameters in the network extraction at the selected points are 79.9 nm, 48.8 nm, 56.5 nm, 72 nm, and 53 nm, respectively.

diameters in the network extraction. For the points from the internal domain, i.e., D and F, the radius values are 34.9 nm and 26.6 nm in the sphere fitting case; and the diameters are 72 nm and 53 nm in the network extraction. This indicates that computed pore sizes between two algorithms are similar in the internal domain, but the surface pores in the sphere fitting have larger pore sizes.

All the selected points show that the sphere fitting results contain more pore size information. For example, the region where C is in PNM becomes one pore with one pore size, 56.5 nm, but in FM the same region contains pores ranging from 14.3 nm to 63.6 nm.

In general, the inscribed diameter cases show a better match with the sphere fitting results than the equivalent diameter cases. Based on the results from PNM, it is more reasonable to use the inscribed diameter than the equivalent diameter for predicting PSD. For the granular materials (Ti, CL), the PSD results from the sphere fitting and PNM show larger discrepancies than those of fibrous materials. As a result, it appears neither PSD cases in PNM are sufficient to represent PSDs of microstructures. To account for more pore size data in PNM, the throat size information should be incorporated into the PSDs obtained from PNM.

3.2 Mercury intrusion porosimetry

To validate the pore sizes assigned by FCST and the network in PNM, mercury intrusion results obtained numerically are compared to experimental results. The mercury intrusion simulations can be done using the pore sizes obtained from the reconstructed images. To obtain the numerical mercury intrusion results using both PNM and FM, the methods discussed in Sections 2.2.2.1 and 2.3.3 were used, respectively. The cumulative intrusion curve created from PNM is computed based upon the throat sizes in the PNM extracted network. As such, the intrusion simulations in PNM should be understood as intruding pores based on the capillary pressures of the throats connected to the pores.

In the FM algorithm, the intrusion accuracy could vary based on the number of incremental pressure steps between P_{start} and P_{end}; and the number of steps for calculating PSD. Since the FM algorithm uses discrete PSD information, the intrusion simulation is highly dependent on the PSD data. If the number of steps for the PSD calculation is not sufficient, many voxels would have the same radius value, which can lead to a big jump in the cumulative intrusion curve. Due to this reason, the number of steps for computing PSD and saturation are set to 100 to achieve evenly distributed saturation values.

Figure 3.11 shows the simulated mercury distributions at various saturation levels (0.2, 0.4, 0.6, and 0.8) using OpenFCST and OpenPNM for Toray 120C. Partial saturation images are obtained as discussed in Section 2.3.3. Figures 3.11a and 3.11b show the mercury saturation of 0.2. Mercury first intrudes the surface pores in both FM and PNM. As mentioned in Sections 2.2.2 and 2.3.3, mercury intrusion takes place from all 6 faces. This is clearly shown in both methods. At low pressure, larger pores connected to the surface are flooded. As pressure increases, smaller pores connected to either mercury clusters or the surface are flooded, as shown in Figures 3.11c and 3.11d. Finally, at high saturation, Figures 3.11g and 3.11h show that only the smallest internal pores are not filled with mercury. The two methods show similar intrusion trends. The marked regions A and B in Figures 3.11c and 3.11d show a similar growth of intruded area, and point C is newly intruded at the saturation level of 0.4. However, the intrusion patterns are not identical. For example, point D in Figure 3.11f is flooded in PNM at saturation of 0.6. The same region is flooded at saturation of 0.8 in FM. The point D has a radius of 19 μm in both PNM and FM, which corresponds to a capillary pressure of 37 kPa. At the saturation level of 0.6, corresponding capillary pressure is higher than 50 kPa based on the cumulative intrusion curves for Toray 120C, shown in Figure 3.12a. This shows that D is flooded at a higher capillary pressure than that of the corresponding pore size. In FM, saturation of the point D is even slower than PNM. This is because mercury clusters connected to D are small pores that require higher capillary pressures to be intruded. The point E in Figure 3.11h shows non-flooded regions at 0.8 saturation in PNM. These pores have pore radii ranging from 14.5 μm to 21 μm, which have capillary pressures between 35 kPa and 50 kPa. As the corresponding capillary pressure at 0.8 saturation is about 65 kPa, the pore volumes in E are going to be a portion of pressures higher than 65 kPa. The discrepancies observed may increase inaccuracy of predicting pore sizes. The effect of larger pores concealed by smaller pores due to their capillary pressures is called shielding effect [111].

Figure 3.12a shows the experimental and numerical mercury cumulative intrusion curves for Toray 120C. Since experimental MIP data for Toray 120C is not available, the experimental data for Toray 090 with 10% PTFE content is shown in Figure 3.12a for reference. Toray 090C has a similar microstructure and mechanical properties compared to Toray 120C, although it has a lower thickness.

Looking at the cumulative intrusion curves for Toray 120C, the experimental curve shows a faster intrusion than the simulated cases. As mentioned in Section 3.1.1, the surface region of the Toray 120C sample is cropped. Since this region with larger pores has been removed from the image, the simulation curve starts to intrude mercury at

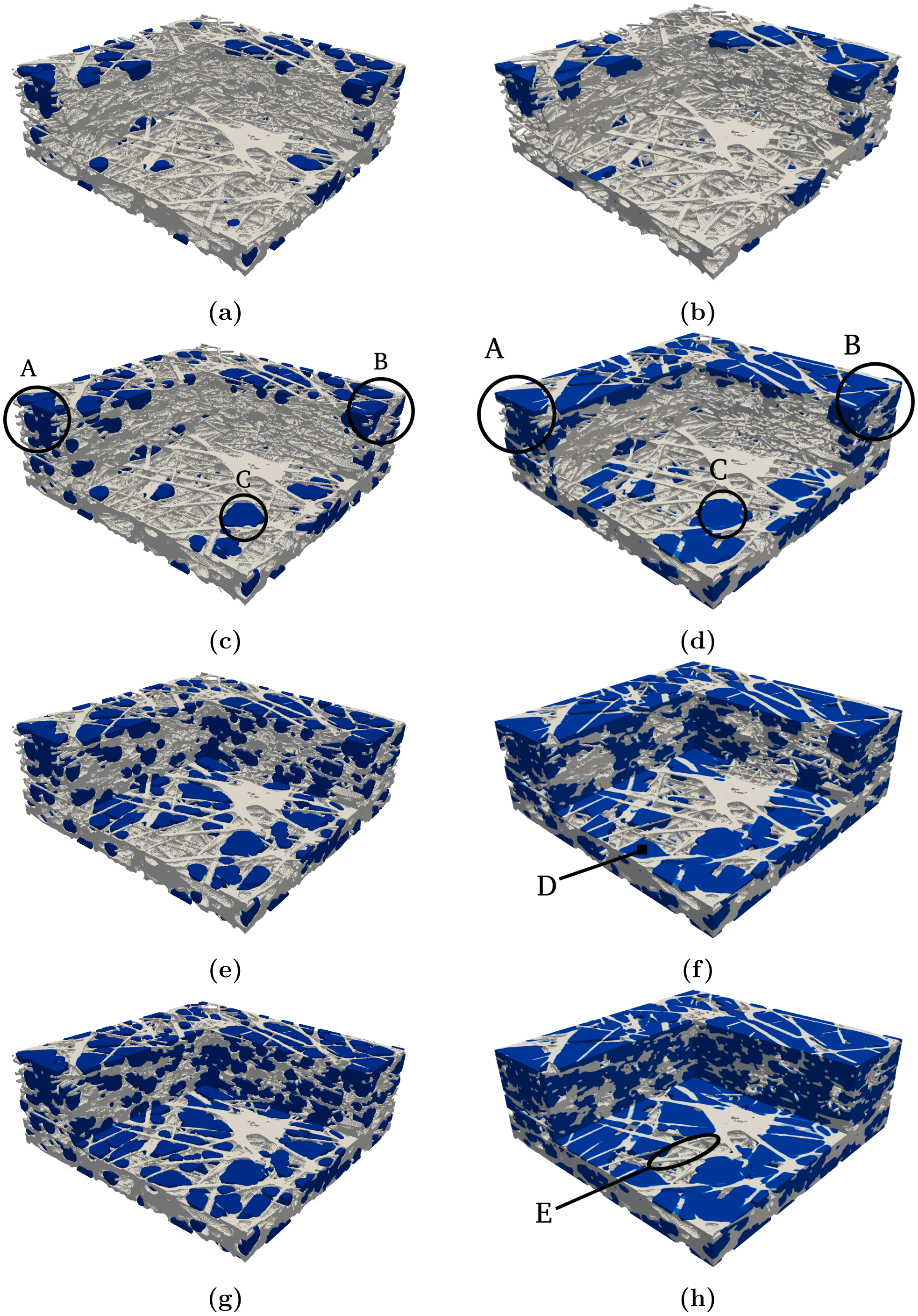
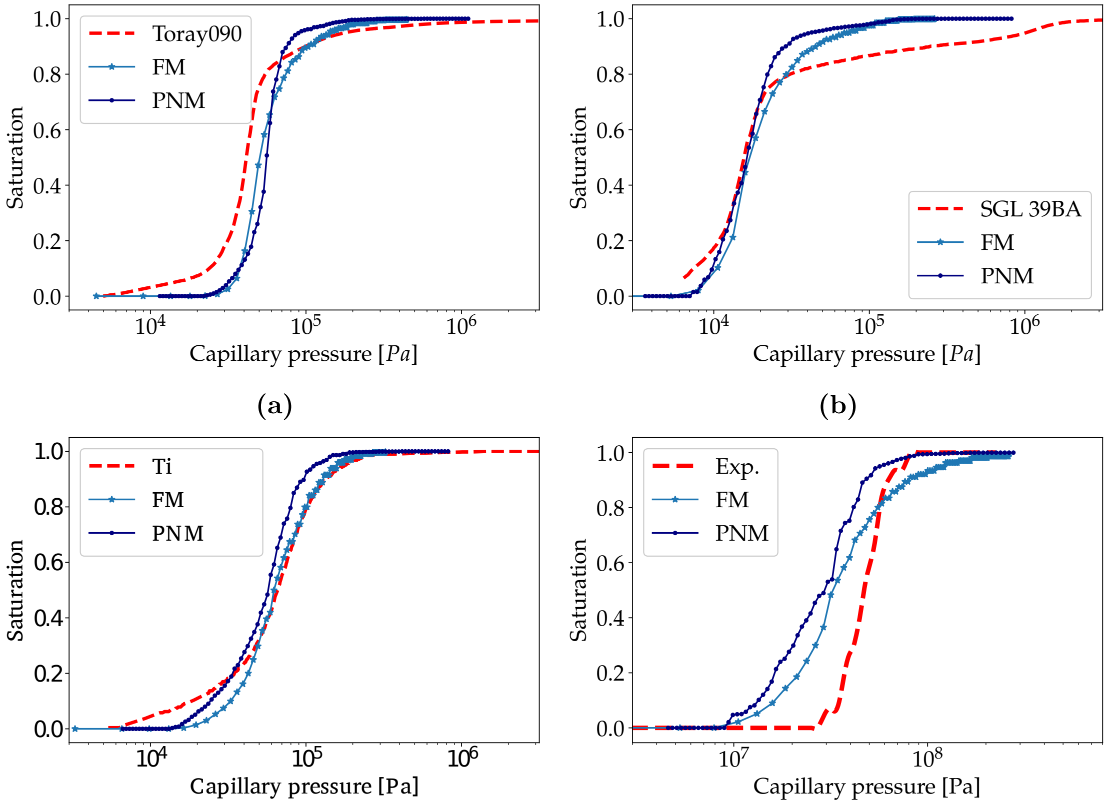


Figure 3.11 – Mercury distribution in partially saturated Toray 120C with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.4 from FM, d) 0.4 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.



(a) **(b)**

(c) **(d)**

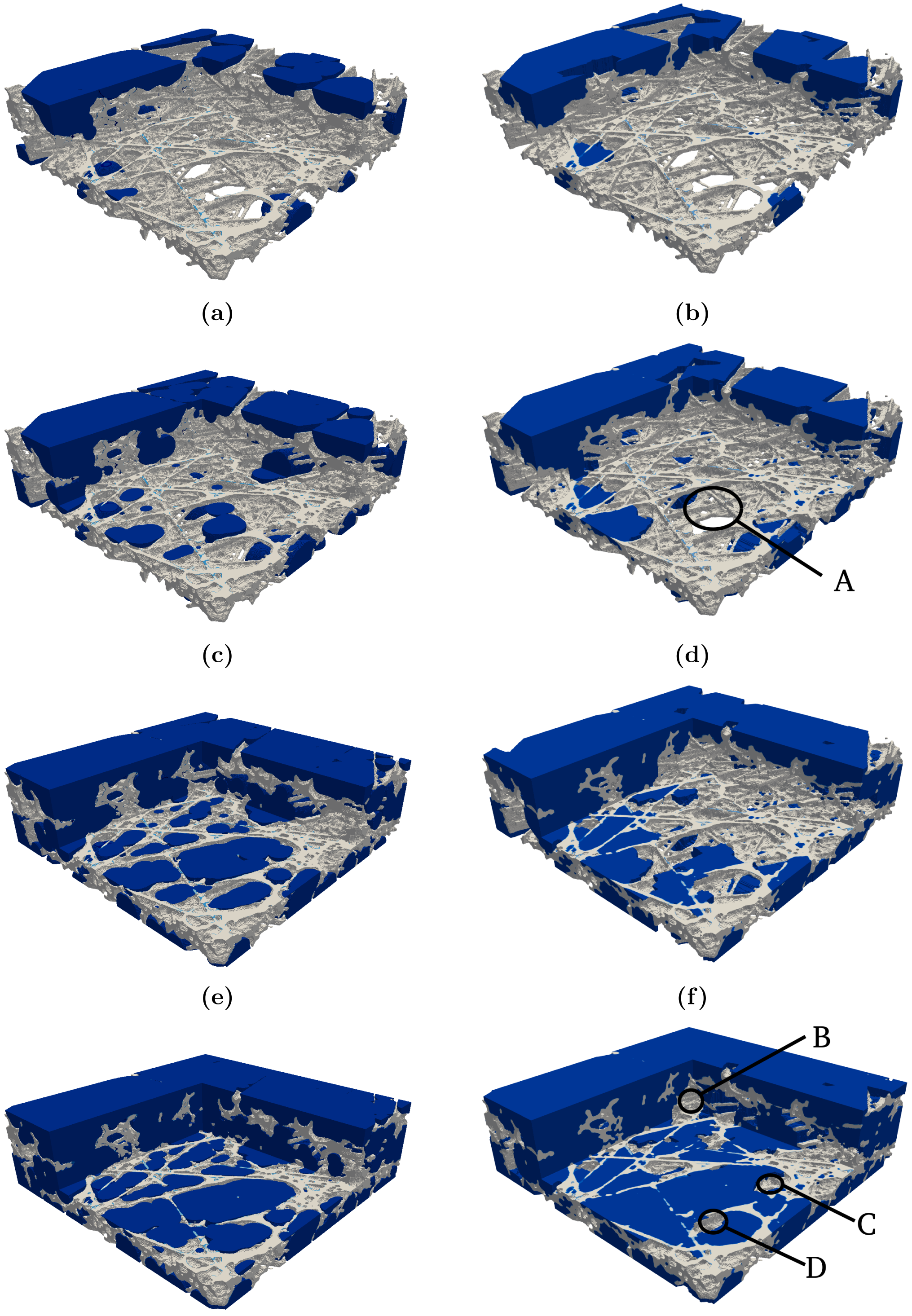
Figure 3.12 – Comparison of numerically predicted mercury intrusion porosimetry results for a) Toray 120C, b) SGL 39BA, c) Ti, and d) CL reconstruction. Each sample is compared to the experimental intrusion curve. For the Toray 120C, the experimental result of Toray 090C is used.

smaller pore sizes. Therefore, it is likely that if the full domain (surface region + core region) was used for the simulations, the virtual MIP would show more portions in the low pressure range. When comparing FM and PNM, the results are similar up to a saturation of 0.7, then the discrepancy increases. The FM curve shows a slower intrusion trend than the PNM curve. This is due to the fact that the FM_PSD was able to capture smaller pores, as assumed in Section 3.1.3.1. This trend is in better agreement with the experimental case.

The mercury distributions in 39BA at various levels of saturation are shown in Figure 3.13. The main difference between 39BA and Toray 120C distributions is because of the effect of the surface regions. Unlike the Toray 120C sample, the surface regions of 39BA are not cropped. Due to this, Figure 3.13a shows that both the top and bottom surfaces are flooded first in both FM and PNM. The mercury region grows internally after the surfaces are sufficiently flooded, which means the surface contains larger pores than the internal part of the material. In fibrous media, cropping the top and bottom surfaces can remove large pores, and this can lead to lowering transport properties. Pore region A in Figure 3.13d is not flooded at saturation of 0.2 in PNM. This region contains three pores with pore radii ranging from 35 μm to 44μm and they are flooded at pressures above 17 kPa. The same region as A in FM has pore radii between 32 μm and 51 μm. The larger size pores in A made flooding easier in FM than PNM. Based on Figure 3.12b, saturation of 0.8 requires 22 kPa in PNM. The maximum pore radii at points B,C, and D in Figure 3.13h are 27 μm, 35 μm, and 26 μm, respectively. These pores are all intruded after the capillary pressure of 24 kPa, which is saturation of 0.86.

For the 39BA sample, neither simulation method could adequately capture the small pores in the carbon matrix (below CT resolution). As a result, the simulated intrusion curves show excellent agreement up to 0.8 (where all pores are captured), but faster saturation compared to the experimental MIP data at the respective pressures which are required to intrude the pores below the voxel size. The voxel resolution of the image stack is 1.79 μm, and the carbon matrix region is assumed as solid, as discussed in Section 3.1.1. PNM and FM show very similar results at low saturations. At high saturations, FM is better able to capture the slow intrusion despite the missing carbon matrix pores.

For the mercury distributions in the Ti samples, Figures 3.14a and 3.14b show intrusion results at the saturation level of 0.2 from FM and PNM, respectively. Both simulation results show similar patterns of intruded area. The intruded regions grow similarly in both methods as the saturation level increases. The discrepancies at the saturation of 0.4 between Figures 3.14c and 3.14d are very small. Figures 3.14e



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.13 – Mercury distribution in partially saturated 39BA with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.4 from FM, d) 0.4 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.

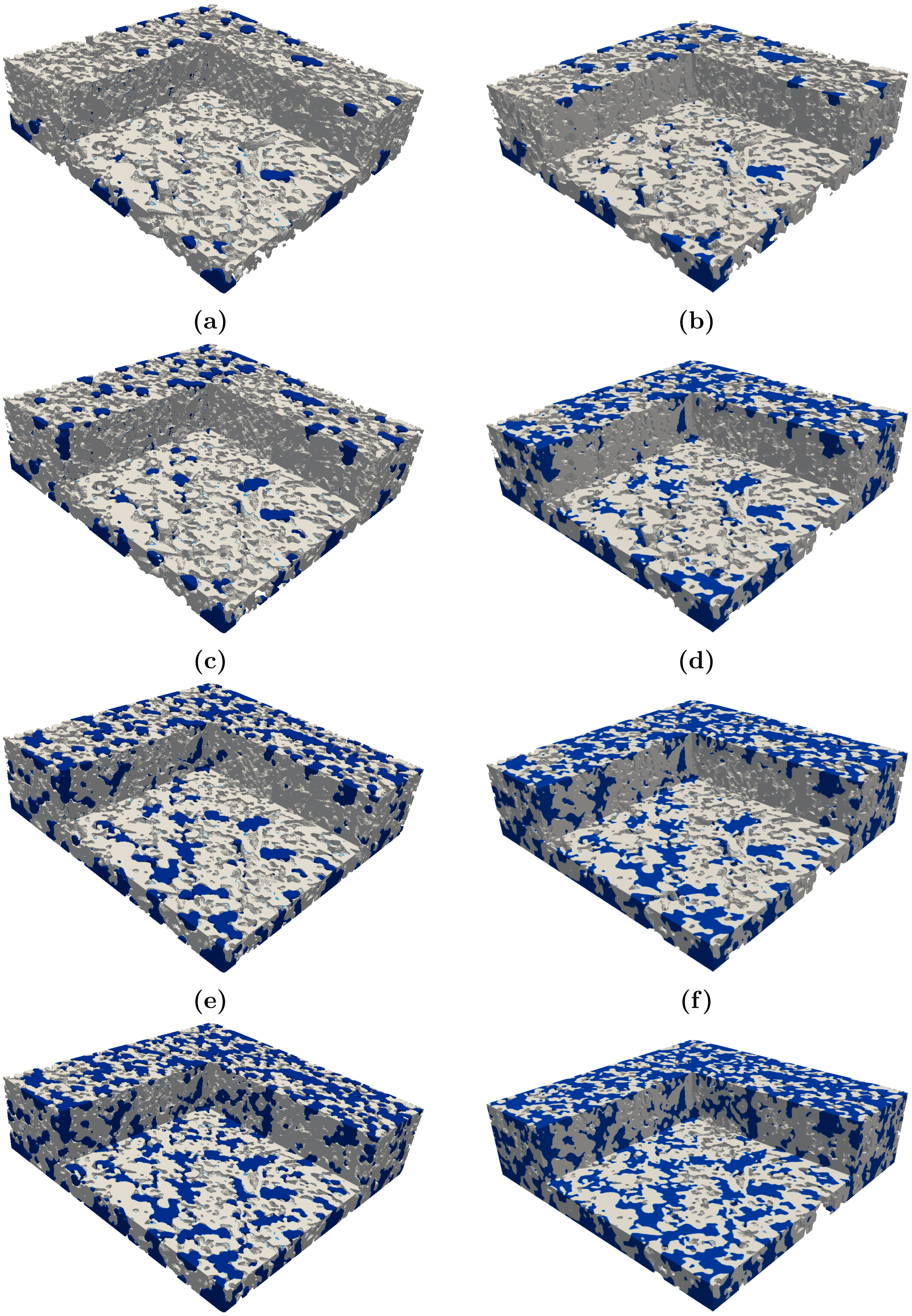
and 3.14f show the mercury saturation at 0.6 where the two algorithms also show similar saturated clusters. The same trends are shown in Figures 3.14g and 3.14h. This implies that, even though the pore sizes computed from FM and PNM are not exactly the same, the intrusion algorithms from FM and PNM work similarly for granular materials. However, as FM has more detail information of pore sizes, intrusion follows a more realistic pattern, as observed in the cumulative intrusion curve in Figure 3.12c.

The intrusion curves in Figure 3.12c show better predictions with FM than with PNM at high capillary pressure (small pore sizes). It can be seen that the FM intrusion curve follows the experimental results closely above a capillary pressure of 50 kPa. Inter-sample intrusion might be responsible for the discrepancy below 50 kPa. The PNM curve has a similar trend, but the intrusion occurred slightly faster than the experimental and FM results.

The CL results for partial saturation are shown in Figure 3.15. The saturation images show similar mercury distributions between FM and PNM. They start intruding the surface of the material, shown in Figures 3.15a and 3.15b, and then proceed to the internal regions (see Figures 3.15e and 3.15f). Finally, nearly all the pores in the cropped part are intruded in FM. In FM at saturation of 0.8, only small pores are left to be intruded, as shown in Figure 3.15g. A and B in Figure 3.15h show discrepancies in PNM from FM at saturation of 0.8. The largest pore radii of A and B are 24 nm and 22 nm, respectively. The fact that these pore regions are not flooded indicates that throats connected to these pores are constricted. This might be a reason for the discrepancy in saturated volume fraction at small pore sizes in Figure 3.12d. For the granular materials, the discrepancies in pore sizes between the surface and the internal regions are not as large as in the fibrous materials.

The cumulative intrusion curves, shown in Figure 3.12d, are compared to the experimental MIP data. The numerical intrusion curves were first obtained by intruding mercury from 6 faces. The CL results show similar trends to the Ti case, in which the FM result shows slower intrusion than that of PNM due to the higher resolution of smaller pores. The experimental result is obtained from the CL samples with porosity of $38.7 \pm 6.7\%$. The simulation results show faster intrusion in general than the experimental data. The experimental MIP curve starts intruding at 20 MPa and reaches full saturation at 90 MPa. The simulation results show large discrepancies compared to the experimental data.

The observed discrepancies might be due to the boundary conditions used for intrusion. CLs are usually very thin ($\sim 5\ \mu$m) and cannot be placed in the penetrometer cell without backing substrates. Due to the backing substrates, only one side of



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.14 – Mercury distribution in partially saturated Ti with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.4 from FM, d) 0.4 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.

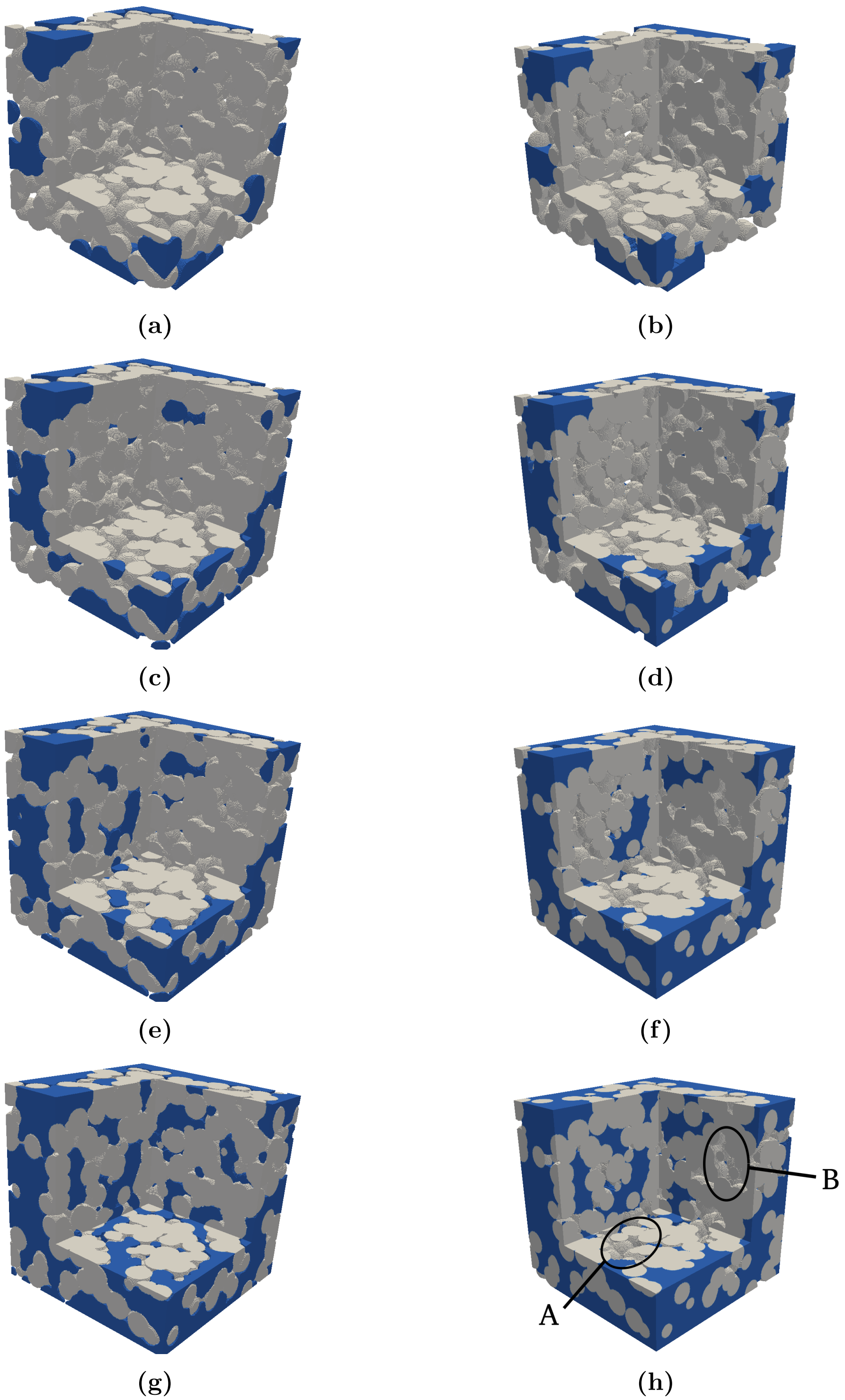


Figure 3.15 – Mercury distribution in partially saturated CL with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.4 from FM, d) 0.4 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.

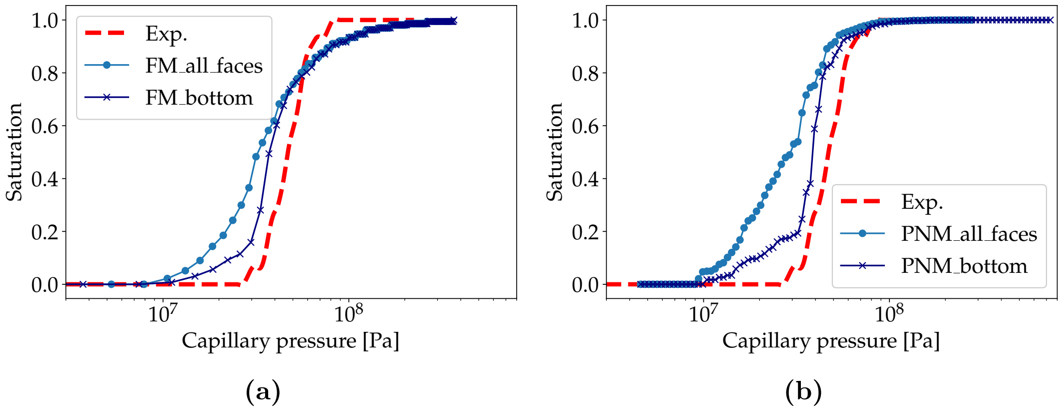


Figure 3.16 – Comparison of the cumulative intrusion curves for CL from a) FM and b) PNM. Mercury is intruded from the bottom face.

the CL can be used as an intrusion boundary. To study the effect of the boundary conditions and assess if they are responsible for the observed discrepancies between simulations and experiments, the simulations were performed such that only the bottom side boundary can be intruded with mercury. Figure 3.16 shows the simulated results for the same stochastic reconstruction sample when it is intruded from one side using FM and PNM. Both methods show slower intrusion than in the previous case. Intrusion starts at a pressure of 10 MPa until a saturation of 0.2. Then, both simulation results show very similar intrusion trends from saturations of 0.2 and 0.6, and the FM result shows smoother saturation after the saturation of 0.6. These simulation results show closer prediction to the experimental data than the results in Figure 3.12d. The experimental result still shows slower intrusion than the PNM simulation, but the intrusion from the bottom face closely predicted the experimental intrusion curves for CL.

As discussed earlier in this chapter, large pores that are accessed through smaller pores are incorrectly included to pressures of the smaller pores. When mercury is intruded from the bottom face, the mercury clusters would have a lower probability of flooding pores at each capillary pressure. Due to this reason, the simulation cases with the bottom boundary in Figure 3.16 show lower saturations than the all-face boundary cases up to 30 MPa. The saturation rate then increases steeply after this pressure. This proves many pores are assigned to incorrect pressures, which correspond to incorrect pore sizes.

In all cases of the cumulative intrusion curves, the FM results show smoother changes when the intrusion curves approach full saturation ($s = 1$). The FM results show better agreement than the PNM results especially for the Ti and CL cases. This is because the FM algorithm segmented the image using a larger number of

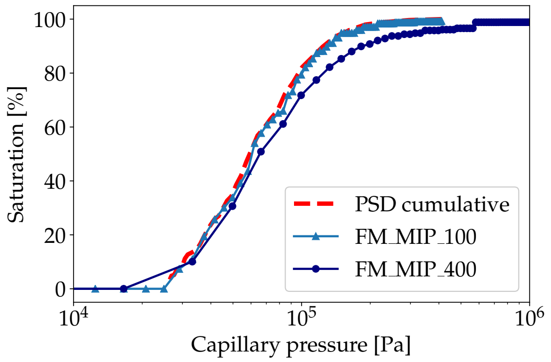


Figure 3.17 – Comparison of cumulative intrusion curves for higher resolution images to an original tomography stack.

small pores than PNM. Both FM and PNM follow analogous intrusion procedures. However, the computed pore sizes by FM and PNM are different (see Section 3.1.3.1), which leads to the different intrusion trends in the same material. The method of sphere fitting in FCST followed by FM is better able to reproduce intrusion trends and therefore is recommended. Based on these results the method should also be able to provide more accurate relative transport properties.

In order to study the effect of the number of voxels, 100 × 100 × 100 voxels of Ti is sampled and enlarged to 400 × 400 × 400 voxels. The original voxel size is 1.79 μm and the refined voxel size is 0.45 μm in all directions. The cumulative intrusion curves for the original images and the refined images are shown in Figure 3.17. The intrusion curve for the original stack is nearly identical to the cumulative PSD curve that the pore sizes converted to corresponding pressures. This explains that the effect of mercury intrusion through constricted parts is very small. The cumulative intrusion curve for the enlarged sample shows lower saturations from 70 kPa, and small portions after 400 kPa. This means the sphere fitting algorithm captured that 4% of the pore regions are assigned to pore radii smaller than 1.79 μm. The discrepancy implies that the voxel size could have an impact on the microstructure analysis. This might need to be studied in future work.

3.2.1 MIP-PSD

Mercury intrusion porosimetry (MIP) is the most widely used experimental technique for estimating pore size distributions (PSDs) of porous media [79, 112, 113]. The PSD can be obtained by first using the Washburn equation (2.12) to relate the fluid intrusion pressure (p_l) to the capillary radius (r_c). The capillary pressure (p_c) is equal to p_l because the sample is under vacuum before intrusion. Then, the cumulative normalized intrusion curve is differentiated with respect to the radius. The logarithmic pore size distribution ($\frac{dX}{d(ln(r))}$) for a given pressure (P_i) normalized with respect to total pore volume (V_{pore}) is calculated using equation (2.12) with the following relations:

$$P \cdot r = -2\gamma cos\theta \tag{3.2}$$

$$dP \cdot r + dr \cdot P = 0 \tag{3.3}$$

$$\frac{dP}{P} = -\frac{dr}{r} \tag{3.4}$$

$$d(ln(P)) = -d(ln(r)) \tag{3.5}$$

$$\frac{dX}{d(ln(r))} = -\frac{dX}{d(ln(P))} = \frac{X_{i-2} - 8X_{i-1} + 8X_{i+1} - X_{i+2}}{12\Delta ln(P)} \tag{3.6}$$

where X_i is the volume fraction of intruded pores with the radius of r_i. The derivative is approximated by a four-point finite difference method [114]:

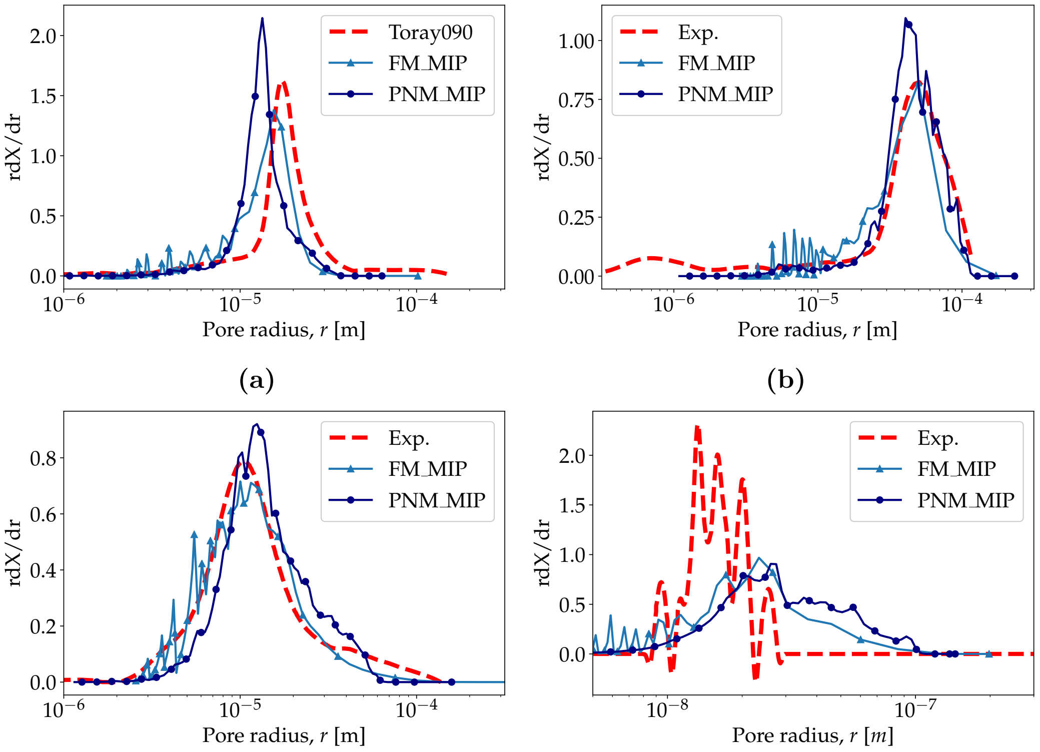
$$f'(x_i) = \frac{f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})}{12\Delta x} \tag{3.7}$$

Since PSD is already defined in Section 3.1.3.1, the pore size distribution obtained by differentiating the cumulative mercury intrusion curve is called MIP-PSD in this work. The primary difference between the MIP-PSD and the PSD is that the MIP-PSD accounts for the connectivity between the pores. The difference between the two kinds of PSDs will be shown in this section.

Figure 3.18 shows the predicted MIP-PSDs for the four samples. Generally, the MIP-PSD from FM is better able to match experiments since the cumulative intrusion is used to calculate PSD. As already mentioned in Section 3.1.3.1, MIP-PSD from FM utilizes a larger number of pores than the PSD obtained from PNM.

The MIP-PSDs of Toray 120C, shown in Figure 3.18a, show that the MIP-PSD from PNM has a higher peak point due to the faster intrusion in the small pores. The experimental PSD is for Toray 090C, which is thinner than the Toray 120C, but has similar bulk density and porosity [80]. The numerically obtained MIP-PSDs have more small pore fractions than the experimental PSD. Based on the MIP-PSDs from FM and PNM, neither result is close to the experimental PSD of Toray 090C. However, considering the experimental data is from a different sample, trends of the numerical and experimental results are similar.

In Figure 3.19a, MIP-PSD from the FM algorithm, FM_MIP, is compared with the PSD from the sphere fitting algorithm, FM_PSD. It is shown that both numerical results have a similar distribution when it comes to the pore size ranges and the probabilities. Therefore, MIP is an effective technique to estimate the PSD of this



(a) **(b)**

(c) **(d)**

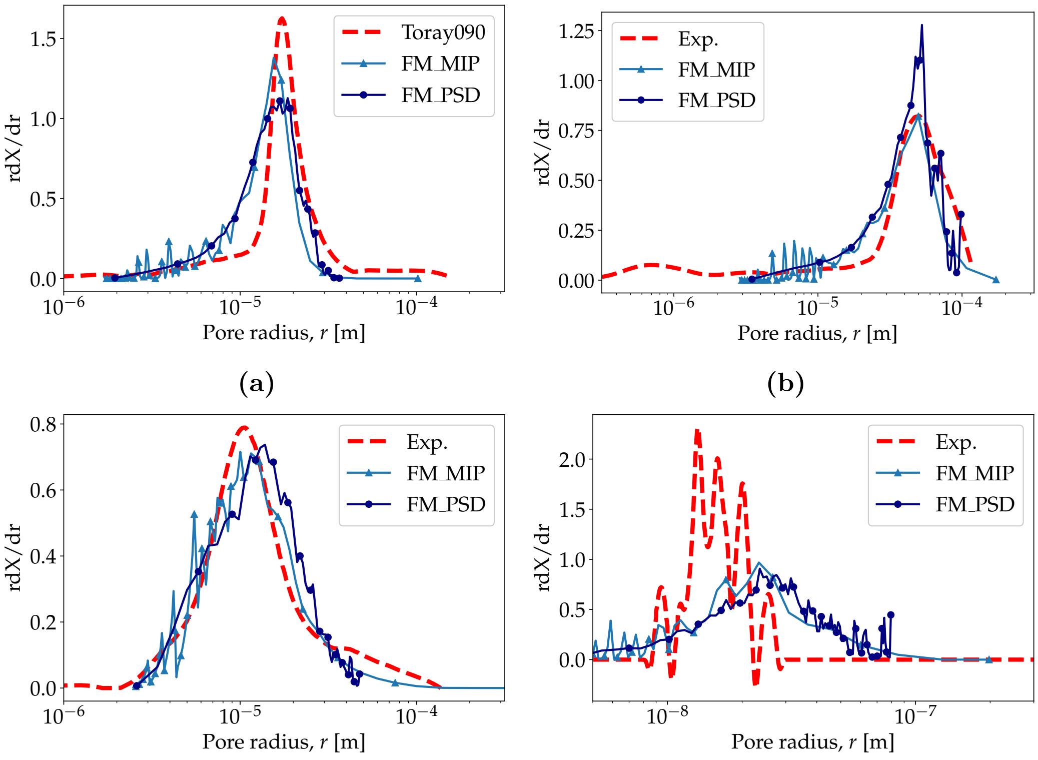
Figure 3.18 – Comparison of numerically predicted MIP-PSD results for a) Toray 120C, b) SGL 39BA, c) Ti, and d) CL obtained from PNM and FM methods.

sample as the intrusion process provides similar results, even though it could result in screening of large pores due to limited connectivity of large pores by small pores.

Figure 3.20a shows the comparison of MIP-PSD and PNM_PSD obtained from PNM. In this case, the PNM_PSD shows similar probabilities to Toray 090 result between 20–30 μm, and have higher probabilities between 6–20 μm. The distribution of PNM_PSD is close to that in the FM results.

Of the four Toray 120C numerical PSD predictions, three cases, i.e., PSD from sphere fitting algorithm, MIP-PSD from FM, and PNM_PSD, show similar distributions. MIP-PSD from FM shows good agreement with the FM_PSD. On the other hand, MIP-PSD from PNM and PNM_PSD show discrepancies to each other. This shows that the shielding effect by small pores in MIP is more significant in PNM for this material.

For the SGL 39BA, the experimental PSD in Figure 3.18b shows a main peak at 40–50 μm and a second smaller peak at 0.7–0.9 μm. Both numerical approaches similarly predicted the main peak. The small pore range around 1 μm, which occupies



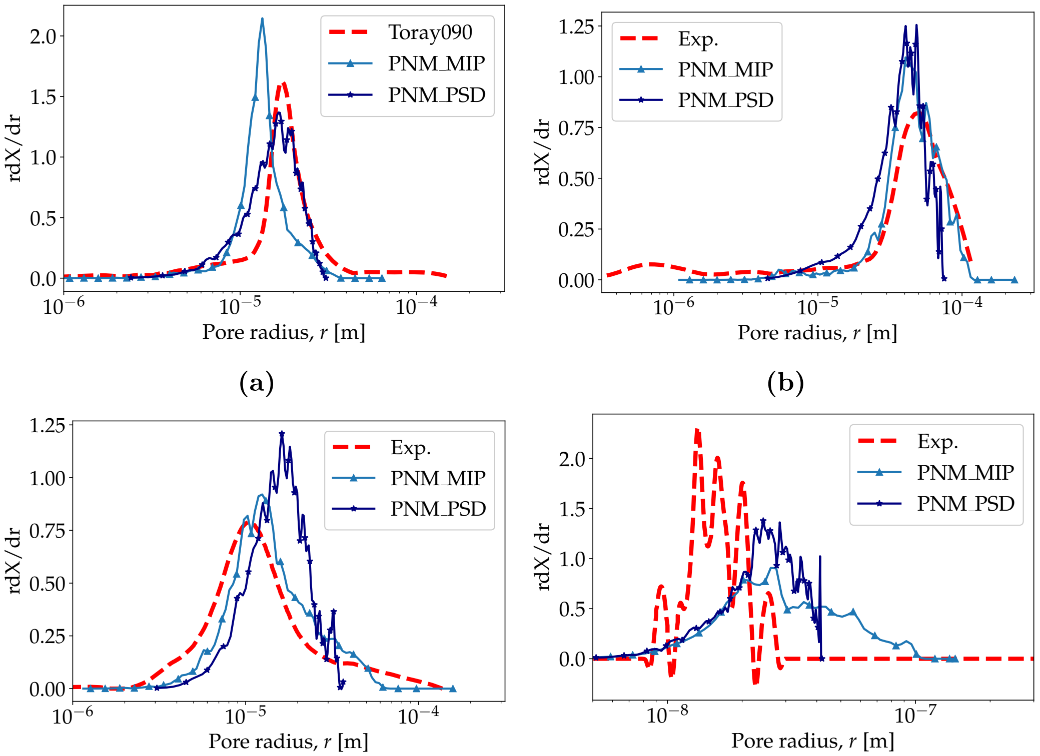
(a) **(b)**

(c) **(d)**

Figure 3.19 – Comparison of numerically predicted PSD and MIP-PSD results obtained by FM algorithm for a) Toray 120C, b) SGL 39BA, c) Ti, and d) CL reconstruction. Each sample is compared to the experimental intrusion curve. For the Toray 120C, the experimental result of Toray 090C is used.

about 9% of the entire pore volume, cannot be observed in the numerically obtained PSDs due to the voxel size. The FM_MIP case contains a higher portion of pores between 10 μm and 30 μm, and lower probabilities in the range of 50–100 μm than the experimental PSD. The pore radii between 5 μm and 10 μm show some probabilities in FM_MIP. FM shows noise within this range. When comparing MIP-PSD from PNM to the main distribution of the experimental PSD, the PNM_MIP case shows a good prediction when estimating pore size ranges. The peak point probability is higher than 1 because the probabilities below the pore radius of 5 μm are nearly zero, and PNM_MIP does not show noise in a small pore range below 20 μm. As the second peak in the experimental PSD of 39BA is unable to be captured in the numerical simulations, the volume fractions of the small pores are treated as larger pores.

The FM_MIP of 39BA shown in Figure 3.19b shows very similar distribution to the FM_PSD result as seen in the Toray case. The FM_PSD has a higher probability



(a) **(b)**

(c) **(d)**

Figure 3.20 – Comparison of numerically predicted PSD and MIP-PSD results obtained from PNM for a) Toray 120C, b) Ti, c) SGL 39BA, and d) CL reconstruction.

at the peak point than the FM_MIP case, since the FM_PSD has no noise in the range of 4–10 μm. For this material, FM_MIP and FM_PSD are in very good agreement for the pore sizes above 50 μm. This indicates that the effect of shielding large pores in MIP is sufficiently small to represent the actual pore size distribution.

The PNM_PSD shown in Figure 3.19d exhibits a similar peak point to the PNM_MIP result, but it does not show large pore portions above 70 μm, and has more small pore probabilities in between 8 μm and 30 μm. This shows that the PNM_PSD, which is computed without considering throat sizes, is not appropriate to represent the actual pore sizes.

For the PSDs of fibrous materials, FM_PSD and FM_MIP show similar results to each other. Even though the FM_MIP results show small discrepancies in large pore ranges and some noise in small pore ranges, the MIP-PSDs from FM are sufficient to represent the actual PSDs. This proves either the sphere fitting algorithm or MIP-PSD from FM can be used to determine PSD of the fibrous materials. This is, however, not the case in PNM. The PNM_MIP for Toray 120C shows a discrepancy from the PNM_PSD. This proves a large amount of pore volume is assigned to incorrect pore sizes due to screened large pores. For 39BA, PNM_MIP shows a better prediction than PNM_PSD because PNM_PSD is not sufficiently represent the PSD of the material. Based on these observations, neither PNM_PSD nor PNM_MIP could successfully represent the actual pore sizes for fibrous materials.

Figure 3.18c provides a comparison of MIP-PSDs for Ti. The FM_MIP case shows better prediction than the PNM result. As the cumulative intrusion curve from FM already showed nearly identical predictions at high pressure ranges, the PSD result obtained from the cumulative curve is also in great agreement. The PNM_MIP case also shows reasonable prediction, but not as good as the FM_MIP.

Figure 3.19c shows the comparison of FM_MIP to the FM_PSD from the sphere fitting algorithm. The FM_MIP and FM_PSD show similar distributions as in the fibrous materials. There are some discrepancies but none are significant. As a result, either of them is appropriate to define the pore size distribution of the material.

Figure 3.20c shows the discrepancies of PNM_MIP and PNM_PSD cases to the experimental result. The PNM_PSD and PNM_MIP show a similar lower limit to each other, but PNM_PSD does not show pores larger than 40 μm. For this material, PNM_MIP shows a better prediction than PNM_PSD, as the PNM_PSD is incorrect due to its insufficient pore sizes. As the PNM_MIP obtained based on throat sizes shows better predictions, the throat sizes should be considered for PNM_PSD as well.

The MIP-PSDs for the CL reconstruction are shown in Figure 3.18d. The results from FM and PNM have similar distributions to each other. The MIP-PSDs obtained from PNM and FM show similar trends as in the Ti case. FM_PSD and FM_MIP show nearly identical distributions as shown in Figure 3.19d, and PNM_MIP has a closer distribution to the results from FM, shown in Figure 3.20d. As a result, MIP-PSD is able to represent the actual pore size distribution for the granular materials. The MIP-PSD results for CL, however, show large discrepancies to the experimental data, because the MIP-PSDs are obtained from the case that mercury is intruded from all faces.

The mercury intrusion of CL from the bottom boundary showed closer intrusion curves to the experimental data than the case using 6 faces in Section 3.2. To compare the MIP-PSDs of the bottom inlet boundary cases, the results for the bottom boundary case are provided in Figure 3.21. The plots exhibit that the MIP-PSD results for the bottom boundary case show closer predictions to the experimental data than the case using 6 faces.

Figure 3.22 shows that the differences in mercury cumulative intrusion curves, and the PSDs obtained from the intrusion curves, depends on the inlet boundaries. As

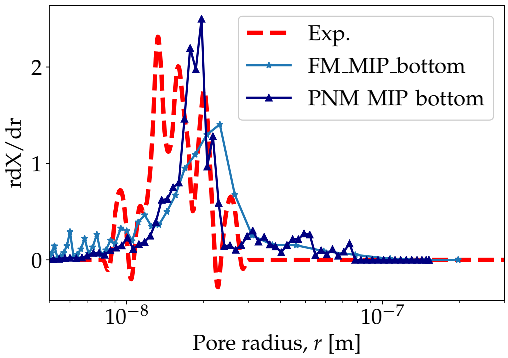


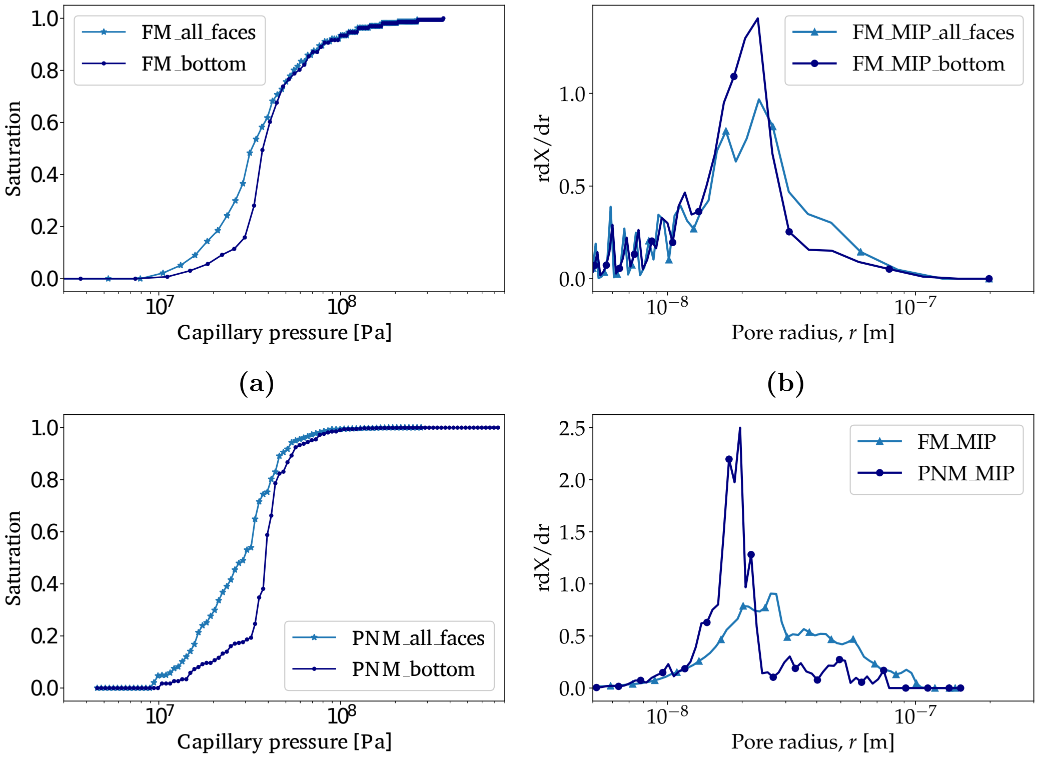
Figure 3.21 – MIP-PSDs for the CL stochastic reconstructions predicted based on the cumulative intrusion curves using the bottom boundary face.

already mentioned in Section 3.2, the intrusion curves, shown in Figures 3.22a and 3.22c, show slower intrusion trends when mercury intruded from the bottom face. The saturation levels in the bottom boundary cases increase steeper than the 6-face cases. This clearly shows that more pore volumes are assigned to higher pressures than the pressures corresponding their pore sizes in the bottom boundary cases. That being said, more pores would be assigned to incorrect pore sizes due to a higher effect of shielding large pores.

Figures 3.22b and 3.22d show MIP-PSDs of CL obtained from the intrusion curves. MIP-PSDs with the bottom boundary show lowered large pore probabilities and raised mid-range pore probabilities in both FM and PNM. The discrepancies between the two boundary cases is larger in PNM than FM due to less connectivity. The experimental MIP for a real CL material is performed with a substrate on one side. This implies the experimental PSD for CL could have incorrectly assigned pore sizes.

For the granular materials, such as Ti and CL, the PNM_PSD results could not show large pore regions. The MIP-PSDs from both PNM and FM show similar distributions to the pore size distributions from the sphere fitting algorithm.

In general, the FM approach shows that PSDs predicted using mercury intrusion cumulative curves are in great agreement with PSDs obtained directly from images. For the PNM method, PSDs obtained from the MIP simulations show better agreement than the pore inscribed diameter cases due to throat size information. However, MIP-PSD from PNM could not always show good prediction of the actual pore size distribution. MIP-PSDs obtained from the intrusion curves using 6-face boundaries are much closer than the single face boundary simulations, as the pore connectivity to the mercury clusters is limited.



(a) (b)

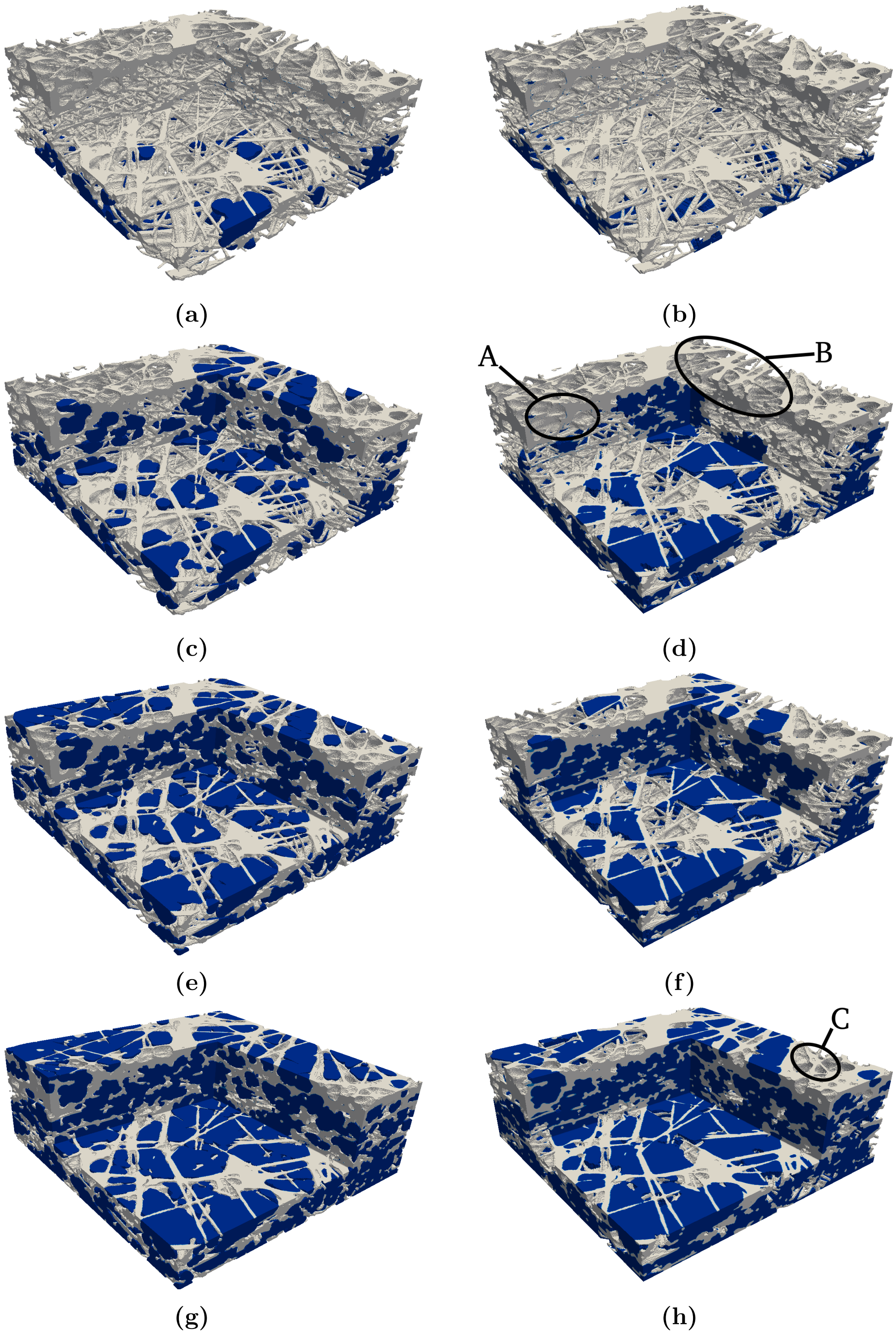
(c) (d)

Figure 3.22 – Comparison of numerically predicted cumulative intrusion curves and MIP-PSDs for CL with the two boundary options; a) cumulative intrusion curves from FM b) MIP-PSDs from FM, c) cumulative intrusion curves from PNM, and d) MIP-PSDs from PNM.

3.3 Liquid water intrusion

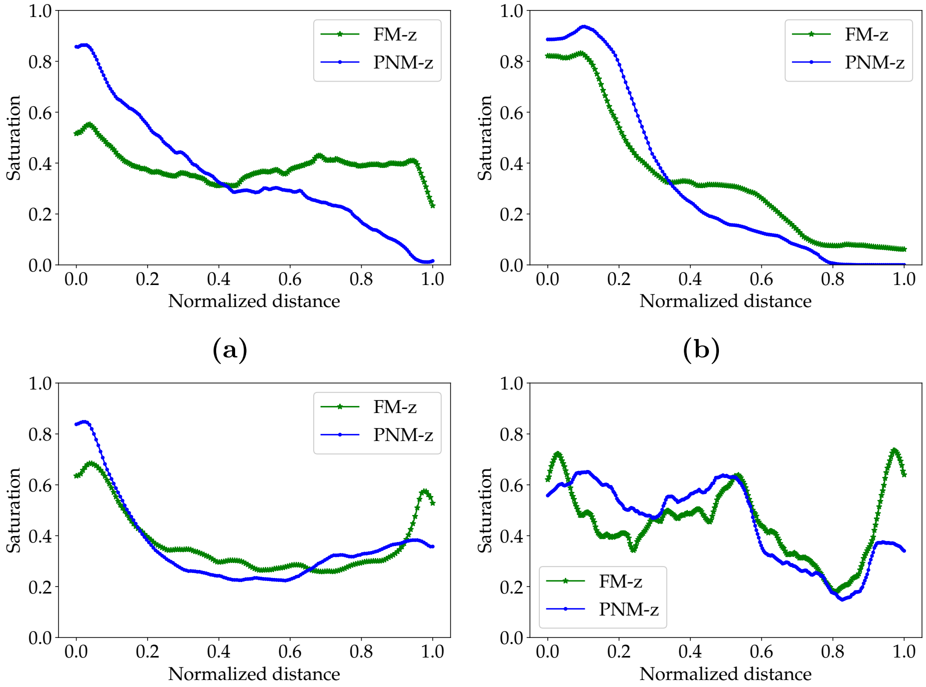
Regarding the liquid water injection in PTLs, water is intruded from the bottom of the domain. In the CL, the ORR takes place on the Pt surface, therefore, the nucleation mode of water injection described in Section 2.3.3 is deemed to be representative of the water intrusion in the CL [54]. In the present study, pores with the smallest radius (2 nm) were selected as the water nucleation points, and used as nucleation sites for the water injection simulations for both FM and PNM. As mentioned in 2.2.2, PNM provides two intrusion algorithms, i.e., invasion percolation and ordinary percolation, and both algorithms are used.

Figure 3.23 shows partially wet images of Toray 120C from FM (left) and PNM (right). The results from both methods show general agreement in propagation of water clusters. Figures 3.23a and 3.23b show the liquid water distributions at the saturation of 0.2, intruded from the bottom face of Toray 120C. The lower left and right faces are intruded in the figures, which proves the water intrusion started from



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.23 – Liquid water distribution in partially saturated Toray 120C with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.38 from FM, d) 0.38 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.



(a) **(b)** **(c)** **(d)**

Figure 3.24 – Through-plane (z-direction) local saturation profiles from the FM and PNM simulations for each sample. The normalized distance 0 is for the inlet face, and 1 is for the outlet face. Each figures shows the local saturation profiles of; a) Toray 120C at 38% saturation; b) SGL 39BA at 35% saturation; c) Ti at 35% saturation; and d) CL reconstruction at 45% saturation.

the bottom face. The FM case shows more water clusters in the surface of the cropped region, but in general both PNM and FM show similar water clusters. Figures 3.23c and 3.23d show the water distribution at the saturation of 0.38 predicted by FM and PNM, respectively. Some discrepancies can be observed. Figure 3.23d shows no water clusters on the left wall (region A) of the cropped region and the top face (region B), whereas water has intruded these regions in Figure 3.23c. The same regions are intruded at the saturation level of 0.6 in PNM as shown in Figure 3.23f. Considering they are at the same saturation, PNM must have intruded pores somewhere else compared to FM. The discrepancies in pore saturations between the two models are explained using local saturation profiles in the next paragraph.

The local saturation results are obtained from the total saturation of 38%, which corresponds to Figures 3.23c and 3.23d. The PNM results have a steeper variation in the z-direction than the FM result. Since the liquid water is intruded from the bottom face, a higher saturation is shown at the inlet face (distance = 0) for both algorithms, but especially for PNM. Although the total saturation is the same, i.e.

38%, for both algorithms, about 23% of the outlet face (distance = 1) is saturated in FM whereas PNM shows nearly zero saturation. PNM shows higher saturations in the z-direction distance range of 0.0–0.4 and lower saturations in the range of 0.4–1.0 than FM. FM results will therefore predict faster breakthrough than PNM, which implies the inlet pores in FM have more connections with similar sized pores to the outlet pores. The higher saturation region in the PNM results indicate that more pores near the bottom face are intruded than FM. That being said, PNM has more water clusters than FM near the bottom face at the same saturation. In the in-plane directions, shown in Figures 3.25a and 3.25b, the local saturation curves are very similar with only small discrepancies between PNM and FM.

The reason that PNM has more water near the inlet is because PNM assigns one pore size at one pore region. Figure 3.26 shows the first slice of Toray 120C from PNM and FM at the saturations of 0.1 and 0.38. Both methods show water clusters at similar regions, but PNM shows a larger portion of saturated parts. As already discussed in Section 3.1.3.1, one pore region in PNM becomes one pore, but several pore sizes are assigned in FM. For this reason, FM occupies the largest pore in one region first and neighbouring pores are occupied subsequently, whereas the same region will be occupied at once in PNM, because it has only one largest value. This discrepancy leads PNM to have more saturated voxels, which results in different distributions of water clusters at the same saturation.

The experimental breakthrough pressure and saturation are 3–4 kPa and 52–53%, respectively [31]. Computed breakthrough pressure and saturation using FM are 3.3 kPa and 38%. Those from PNM are 4 kPa and 34%. For both cases, the breakthrough pressures are within the experimental range, but the saturation values are lower than the experimental data.

At the saturation level of 0.6, both PNM and FM show similarly distributed water clusters, but FM shows more breakthrough clusters at the top face. Figures 3.23g and 3.23h show the water clusters at saturation of 0.8. Most pores are saturated with water in a similar manner, but pore region C at the top right face still shows discrepancy between PNM and FM. Pores in the region C have sizes ranging from 4 μm to 13 μm in both PNM and FM. Since the pore sizes in both methods are similar, the reason for the late flooding is due to the differences in pore connectivity. Unlike the PNM case, only small pores remain unintruded in FM. As a result, the pores in C would be concealed at higher pressures. This would lead to a discrepancy in the cumulative saturation curves between FM and PNM. Since FM shows more water clusters than PNM at the saturation level of 0.38, it could be assumed that FM reaches the outlet more easily, due to having more pore size information than PNM.

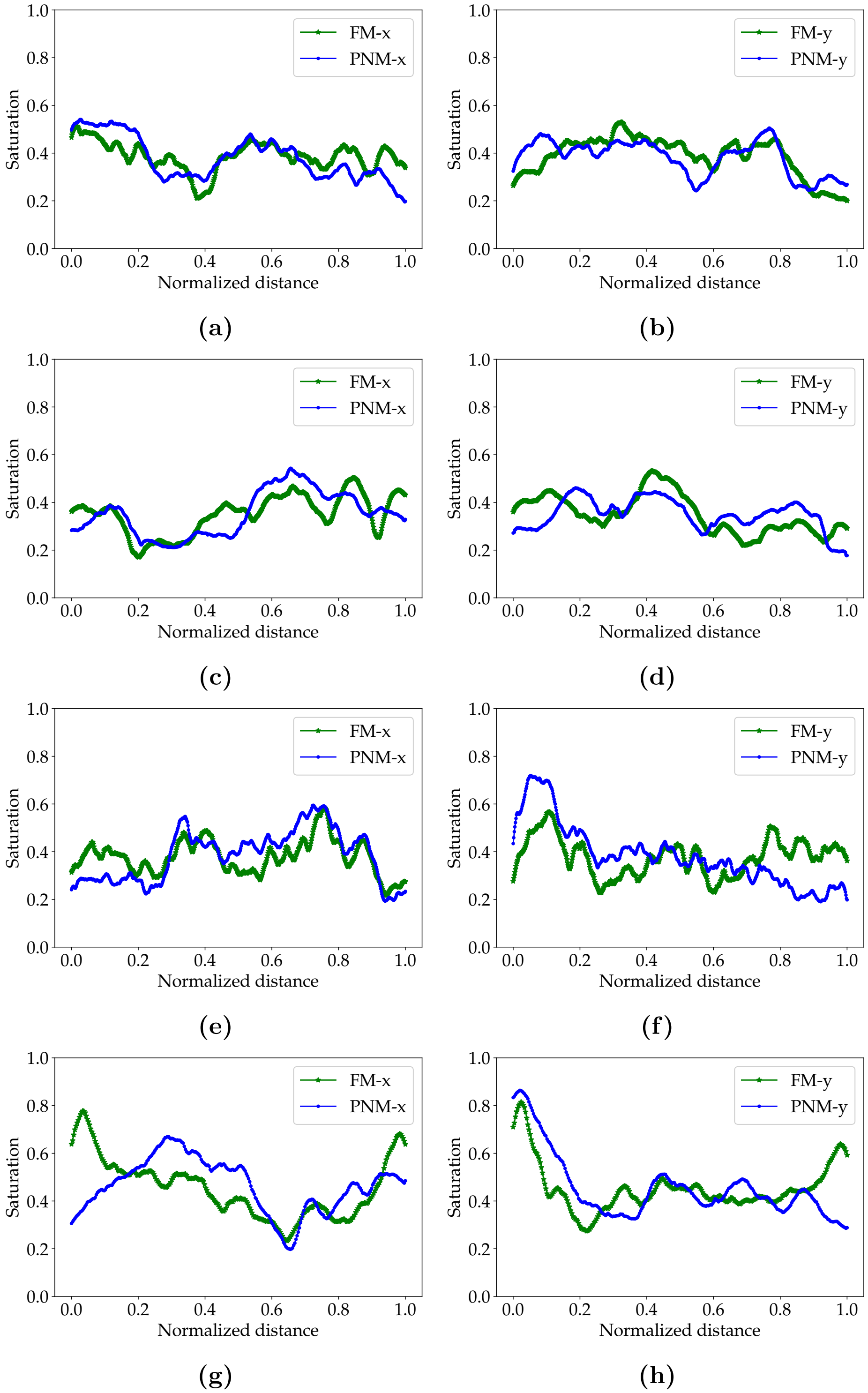
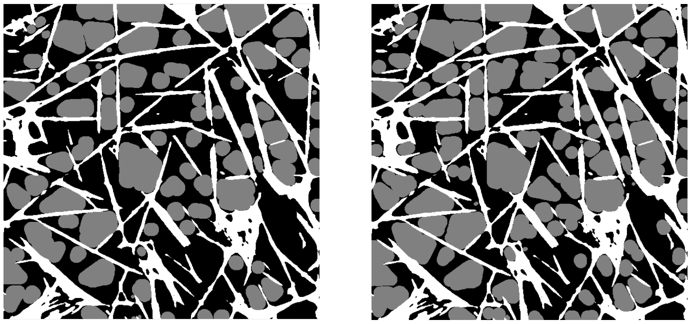
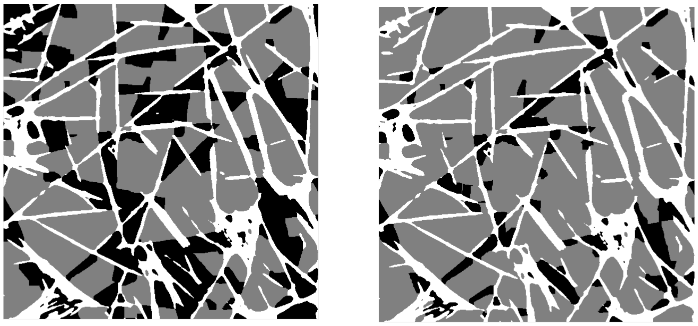


Figure 3.25 – In-plane (x/y-directions) local saturation profiles from the FM and PNM simulations for each sample; Toray 120C at 38% saturation (a) and b)); SGL 39BA at 35% saturation (c) and d)); Ti at 35% saturation e) and f); and CL reconstruction at 45% saturation (g) and h)).

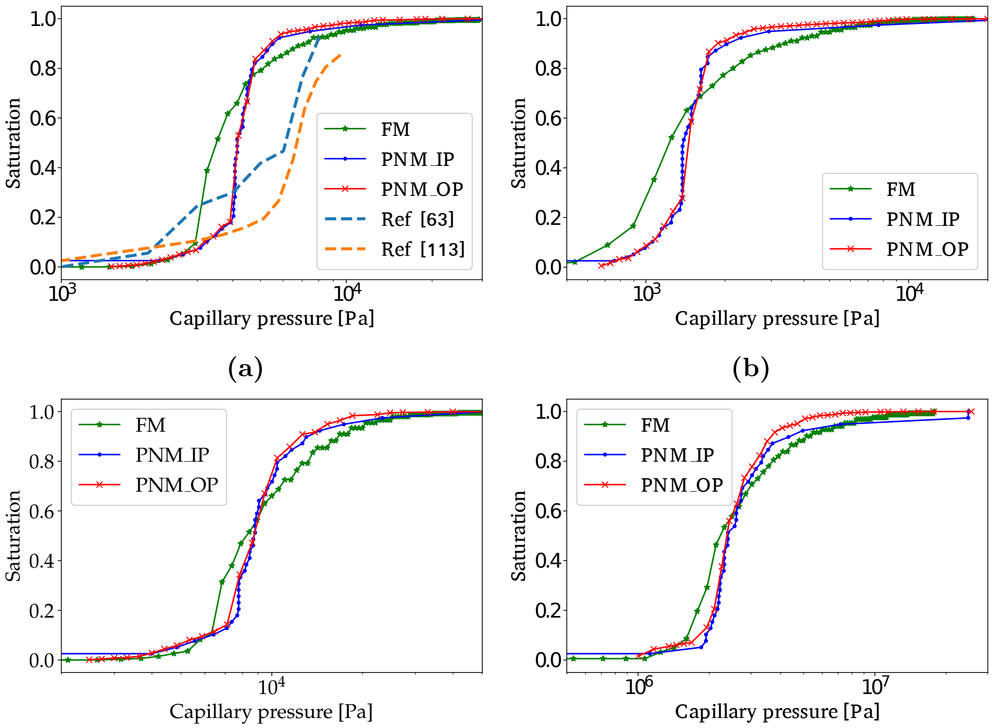


FM sat. = 0.1 FM sat. = 0.38



PNM sat. = 0.1 PNM sat. = 0.38

Figure 3.26 – Comparison of saturation trends between FM and PNM. The water clusters at the bottom slice of Toray 120C at 0.1 and 0.38 saturations where the water clusters are gray, pores are black, and solid regions are white.



(a) (b) (c) (d)

Figure 3.27 – Comparison of numerically predicted water intrusion results for a) Toray 120C, b) SGL 39BA, c) Ti, and d) CL reconstruction.

Comparison of simulated capillary pressure–saturation curves for Toray 120C, shown in Figure 3.27a, shows that the water intrusion in PNM took place rapidly in a saturation range of 0.2–0.9. A reason for this trend is limited information of pores and their connectivity. As observed in Section 3.2, large pores that are intruded through smaller pores are flooded at higher pressures than the pressures to their sizes, which is called shielding effect. The effect of shielding was even larger when using the bottom boundary. Due to this reason, the intrusion curves from FM and PNM show discrepancies. However, as can be seen in Figure 3.27a, both numerical water intrusion results are different from the experimental results of Gostick et al. [115] and Garcia-Salaberri et al. [63]. The discrepancies between experimental and FM results were already shown by Sabharwal et al. [30]. Since the liquid water intrusion in both numerical methods are computed based on capillary pressures of previously determined pore sizes, the result could vary with respect to the inlet pore sizes and contact angle. If the surface area of the Toray image was included, the intrusion would have started from the lower pressure range. Furthermore, if the various contact angles of the Toray material were considered, the numerical intrusion curves would shift to a higher pressure range.

Figure 3.28 shows the water saturation images of 39BA. As in the previous case, there is general agreement between the two models, but there are also some distinct differences. Figures 3.28a and 3.28b show the liquid water intrusion of the 39BA sample from the bottom boundary in FM and PNM, respectively. At this saturation (20%), both results show very similar water clusters to each other. At the saturation of 35%, however, the FM saturation result in Figure 3.28c shows some water clusters at the top, but these are not seen in PNM, shown in Figure 3.28d. This shows that the liquid water intrusion in PNM is slower in penetrating the sample. As in the Toray case, PNM would have more water clusters near the bottom face than FM.

In order to see the discrepancies in water distributions between the two methods, local saturation profiles are computed. The through-plane local saturation profiles for SGL 39BA at 35% saturation are shown in Figure 3.24b. The FM results show a steep decreasing function similar to the PNM results, since the inlet pores are very large compared to the internal pores. The FM profile shows lower saturations at the distance between 0 and 0.3, but it shows higher saturations between 0.4 and 1 than PNM. At the outlet surface, there is no saturated portion in PNM, but FM shows about 8% of saturation. As FM has a higher middle region saturation, it should be more likely to breakthrough than PNM. This explains the discrepancy of the water clusters in Figures 3.28c and 3.28d. PNM could not achieve a breakthrough at the outlet until 90% of the inlet pores are saturated, but more pores near the bottom face are flooded as in the Toray case. High saturation at the inlet and very low saturation at the outlet surface explain that the surface region can have an important role in total saturation rate, which can affect the relative transport properties. In the in-plane case shown in Figures 3.25c and 3.25d, the saturation profiles from both algorithms are in a similar range of saturation.

The experimentally measured breakthrough pressure and saturation values are 1.8 kPa and 25%. These are measured in our laboratory using the SGL 39BA samples. Those values obtained numerically are 1 kPa and 36% in FM, and 1.4 kPa and 45% in PNM. The experimental results require a higher pressure and a lower saturation to reach the breakthrough than the simulations. The missing small pores might attribute to the discrepancies between the experimental and numerical results, since more small pores between large pores would limit pore connections and the water intrusion within the domain.

Figures 3.28e and 3.28f show the water saturation of 60% for FM and PNM. PNM shows no water clusters at region A, but FM started intruding the same region. FM at 0.8 saturation in Figure 3.28g shows more water clusters at the top than PNM in Figure 3.28h, although they are at the same saturation level. The PNM result still does not show the water clusters at the region A. Because each pore in PNM has limited pore connections, the shielding effect in PNM is more significant than FM. Considering the thickness, i.e., 310 μm, and the mean pore size, i.e., 45 μm, of 39BA, less than 8 pores would be placed in the through-plane direction in PNM, which might not be sufficient to depict the water intrusion trends.

The cumulative intrusion curves for 39BA are shown in Figure 3.27c. The numerical intrusion curves show similar trends as in the Toray cases. The FM results show more gradual saturations than PNM. PNM shows a steep increasing function from 0.3 saturation, which indicates a significant shielding effect of larger pores in the domain.

The partially wet figures for Ti are shown in Figure 3.29. As the Ti materials are hydrophilic, the non-wetting phase is air. The water contact angle used in literature is usually 50° [44, 116]. As the intrusion algorithms are built to mimic the intrusion of non-wetting phase, Ti cases in this work show water displaced by air in the domain. Due to this, the blue regions in Figure 3.29 are intruded air clusters in the domain that is fully saturated with water at first.

Overall saturation trends in FM and PNM are similar, as the intruded clusters increase by occupying the same regions. Figures 3.29a and 3.29b shows that only a small amount of the bottom pores are intruded in PNM. On the other hand, at least one pore at the top surface in the FM case is intruded. In the cropped region, many pores that are intruded in FM, are not saturated in PNM. This implies that water clusters near the bottom face would be displaced more in PNM. As the saturation level increases to 0.35, the discrepancy in saturated regions between FM and PNM gets larger. The selected pore regions A in PNM shows no intruded clusters, whereas the same regions in FM have intruded clusters. Point B in Figure 3.29d shows a small air cluster beneath the top face. The water clusters in these regions are displaced at saturation of 0.6 in PNM. In order to find the discrepancies in intruded clusters at saturation of 0.35, saturation at each slice needs to be compared.

Figure 3.24c shows the through-plane saturation profile for Ti. The differences of pore sizes between surface and internal regions are not as significant as in the fibrous material in Ti. Both PNM and FM show some saturations at the outlet. As the inlet pores of Ti are smaller than those of fibrous materials, the number of the inlet pores are sufficient to make connections with similar sized neighbouring pores. This allows the Ti case to have higher saturation at the outlet face. The FM result shows saturation of 0.55 whereas PNM shows a saturation level of 0.35 at the outlet surface. In the FM case, saturation at the outlet is higher than the internal region ($0<\text{distance}<1$). Since the surface has bigger pores than the internal region, once

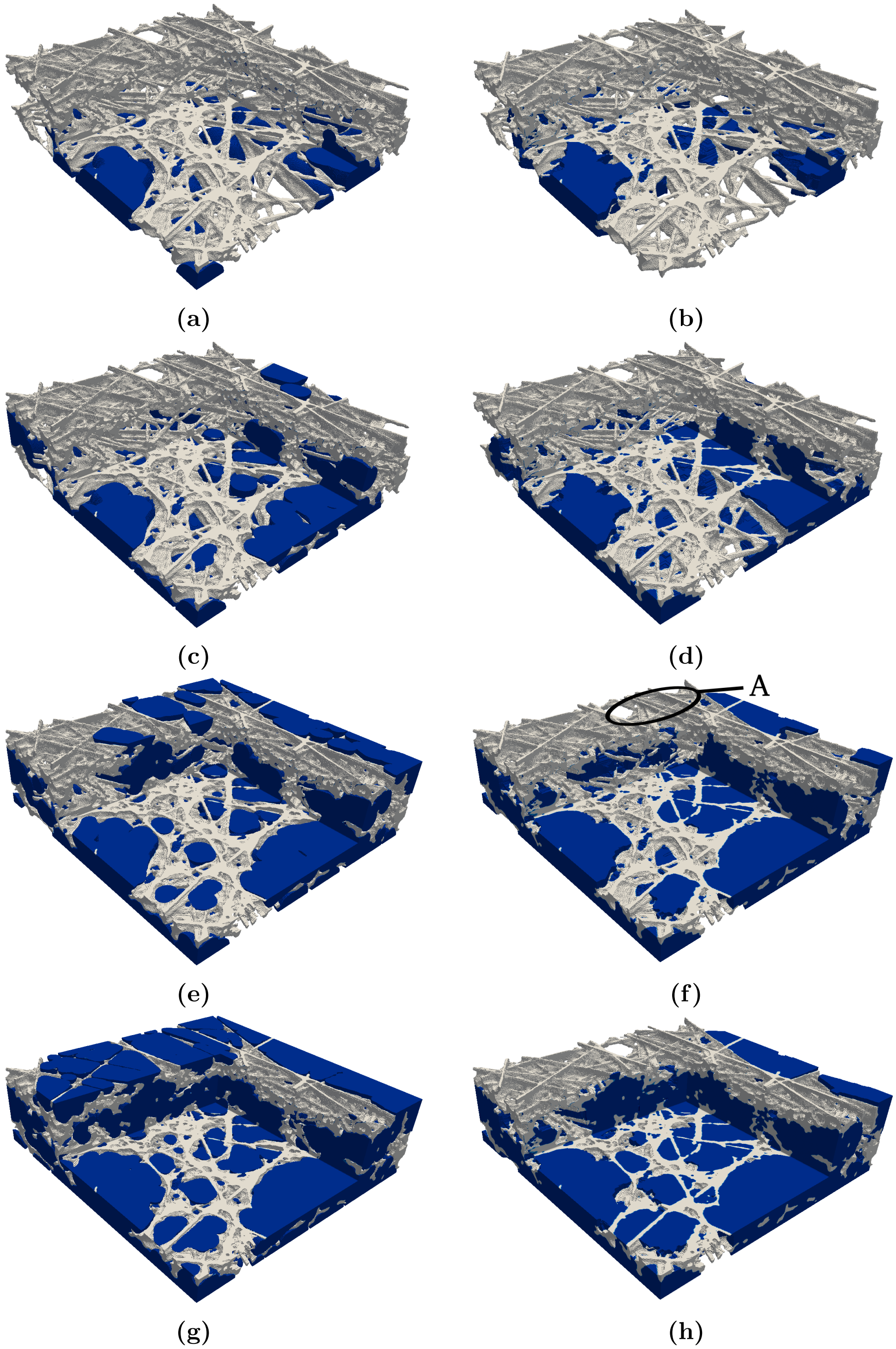


Figure 3.28 – Liquid water distribution in partially saturated 39BA with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.4 from FM, d) 0.4 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.

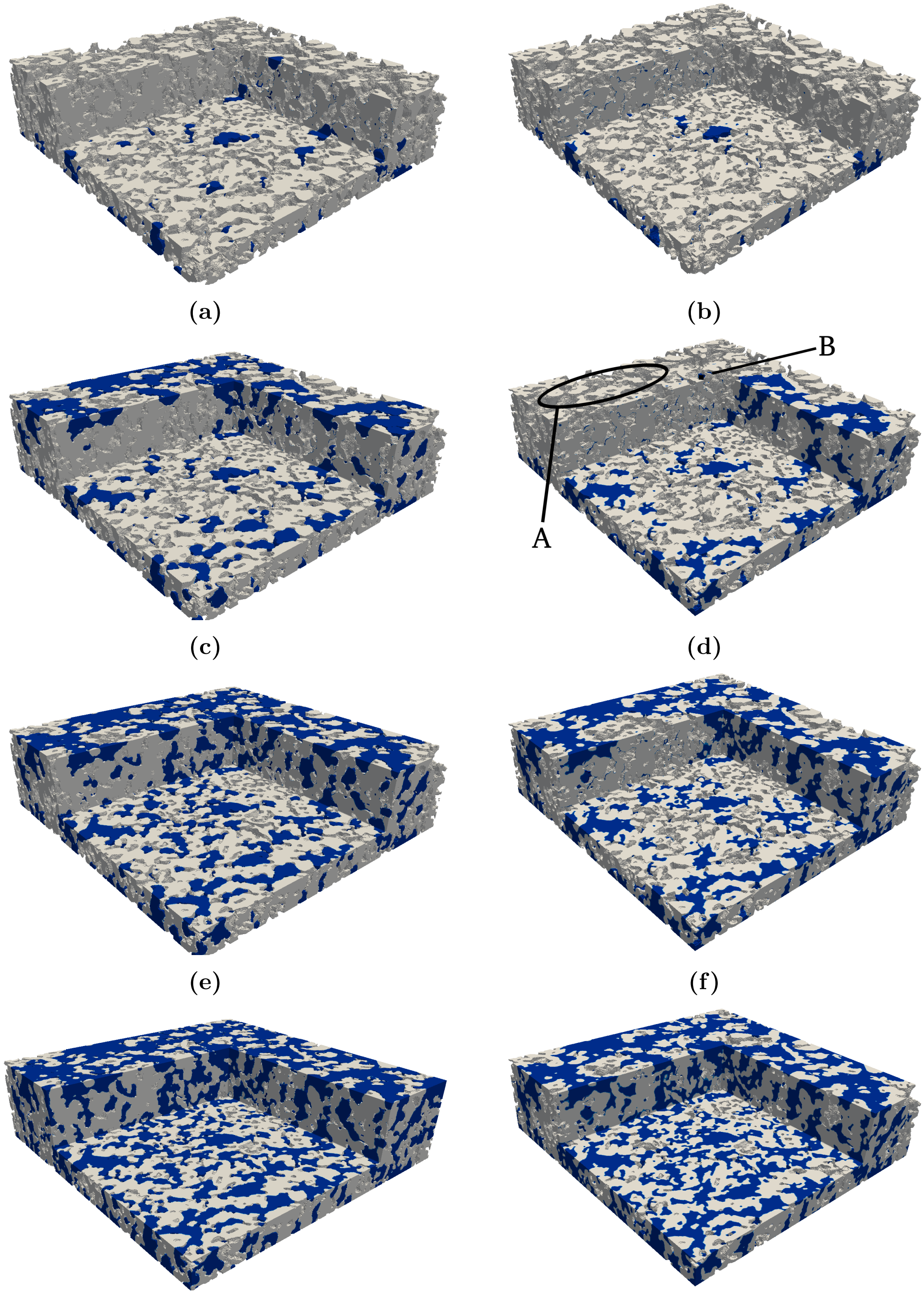
intruded clusters reach a breakthrough to the outlet surface, the saturated portion at the outlet surface could be higher than the internal region. The air clusters occupy more than 80% of the inlet surface in PNM, while the inlet saturation in FM is about 60%. This implies that FM requires less pressure to reach the outlet than PNM. The saturation profiles in the in-plane directions are shown in Figures 3.25e and 3.25f. The local saturation curves of PNM and FM show similar saturation trends in the x-direction. In the y-direction, PNM shows a higher saturation at the distance between 0 and 0.4, and a lower saturation than FM between 0.7 and 1.

When the saturation reaches 0.8, the intruded regions are similar in both methods. Based on the observations, the water displacement in PNM is slower than FM up to 0.6, and then becomes faster after 0.6.

This is proven in the cumulative intrusion curves, shown in Figure 3.27c. The discrepancies between the intrusion curves from PNM and FM are not as significant as in the fibrous materials. This shows that the shielding effect in Ti is less significant than the fibrous material cases.

When the liquid water is displaced by air from the bottom face, the FM results generally show faster breakthrough than the PNM. As discussed already in Section 2.2.2, the intrusion algorithms proceed based on the pore sizes obtained from each algorithm. The more pore information present, the more realistic the intrusion. To account for the gradual intrusion in PNM, OpenPNM provides a function called late pore filling [25, 42]. This function first intrudes 75% of a pore volume at a corresponding intrusion step, and the residual pore volume is saturated at the next intrusion step. However, it could not mimic gradual intrusion procedures successfully.

For the CL stochastic reconstructions, the liquid water saturation at 0.2 are shown in Figures 3.30a and 3.15b. The liquid water occupancy grows from the nucleation sites in the inner domain. As PNM cannot capture all the smallest pores, i.e., 2 nm, that FM can, the nucleation sites would be different between the two algorithms. Even though the liquid water intrusion might be started from different positions in each method, the liquid water clusters show a similar overall growth. At saturation of 0.2, shown in Figures 3.30a and 3.30b, FM shows more water clusters than PNM. The FM result at this saturation have distributed water clusters, as the smallest pores could be anywhere based on the distance map. Water in PNM are all connected to the cluster A in Figure 3.30b. At the saturation level of 0.45, shown in Figures 3.30c and 3.30d, PNM shows similar water clusters to FM in B and C. The water clusters on the left and right edges shown in the FM case are, however, not seen in the PNM figure. In order to understand where the water clusters are placed in PNM, local saturations are discussed in the next paragraph.



(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)** **(h)**

Figure 3.29 – Intruded air distribution in partially saturated Ti with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.35 from FM, d) 0.35 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.

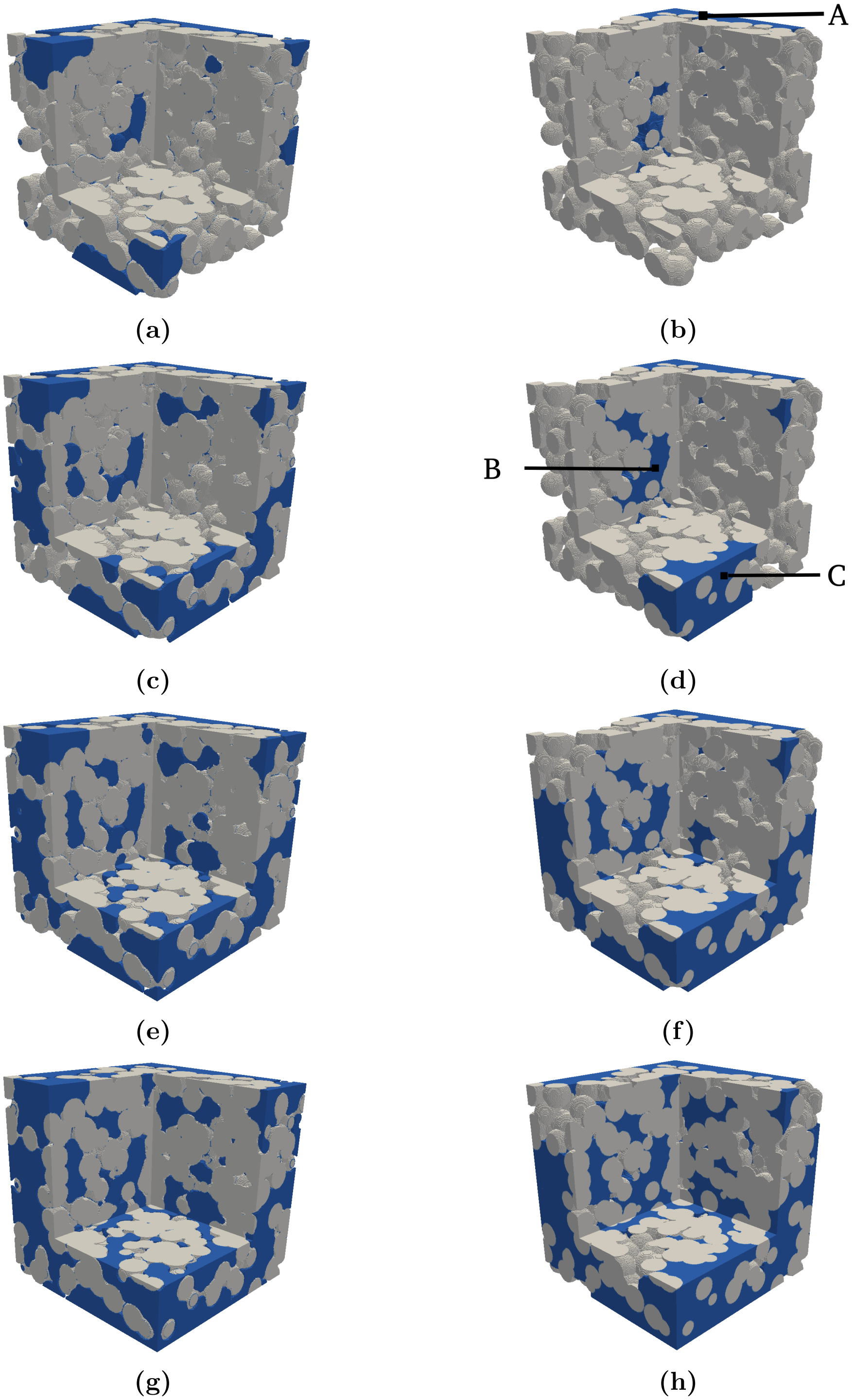


Figure 3.30 – Liquid water distribution in partially saturated CL reconstruction with a saturation level of a) 0.2 from FM, b) 0.2 from PNM, c) 0.45 from FM, d) 0.45 from PNM, e) 0.6 from FM, f) 0.6 from PNM, g) 0.8 from FM, and h) 0.8 from PNM.

The through-plane local saturation profiles for CL at 0.45 saturation are shown in Figure 3.24d. Since the nucleation points of the liquid water intrusion in the CL reconstruction are placed within the domain, the results show larger fluctuations than the other materials. PNM shows higher saturations in the distance between 0.1 and 0.5, which explains more water clusters within this range. FM has higher saturations at the top and bottom faces (distance of 0 and 1, respectively) than the PNM results. This shows that the pore connections in FM have more probabilities to reach the top and bottom pores from the nucleation sites. The two results show similar variations in the distance of 0.3–0.9, which explains that the water intrusion in the internal domain proceeds similarly in both algorithms. In the in-plane directions, shown in Figures 3.25g and 3.25h, the largest discrepancy is seen in the surface of the domain.

Figures 3.30e and 3.30f show the saturation at 0.6. FM starts intruding the bottom side of the cropped region, whereas PNM remains nearly the same as the previous step, i.e., saturation of 0.45. When the saturation is 0.8, shown in Figures 3.30g and 3.30h, the water clusters at the bottom of the cropped region in PNM do not increase as much as in the FM result. As they are at the same saturations, PNM would have saturated pores in different parts of the domain, such as the cropped region.

The cumulative intrusion curves for CL are shown in Figure 3.27d. The PNM and FM results for the CL reconstruction show smoother increasing functions than the fibrous PTLs, which are similar to the Ti cases. As in the Ti case, CL shows that the intrusion in PNM proceeds faster than FM within the saturation range of 0.1–0.9 because of the larger shielding effect in PNM.

3.4 Dry effective diffusivity

Diffusion simulations in PTL and CL domains were performed as discussed in Chapter 2. For the DNS simulations, the average simulation time for the dry PTL samples, with around 50 million DOFs, was around 5 hours per direction and 20 minutes for the CL reconstruction on a single core of Intel(R) Xeon(R) CPU E5-2690 v2 with a clock speed of 3.00 GHz. For the PNM simulations using the same images, the entire simulations in all directions for one image stack took around 5 minutes for PTL samples and 2 minutes for the CL reconstructions. Domain sizes used for the simulations are $600 \times 600 \times \delta_t$ voxels for the GDL and PTL samples, and $300 \times 300 \times 300$ voxels for the CL reconstructions.

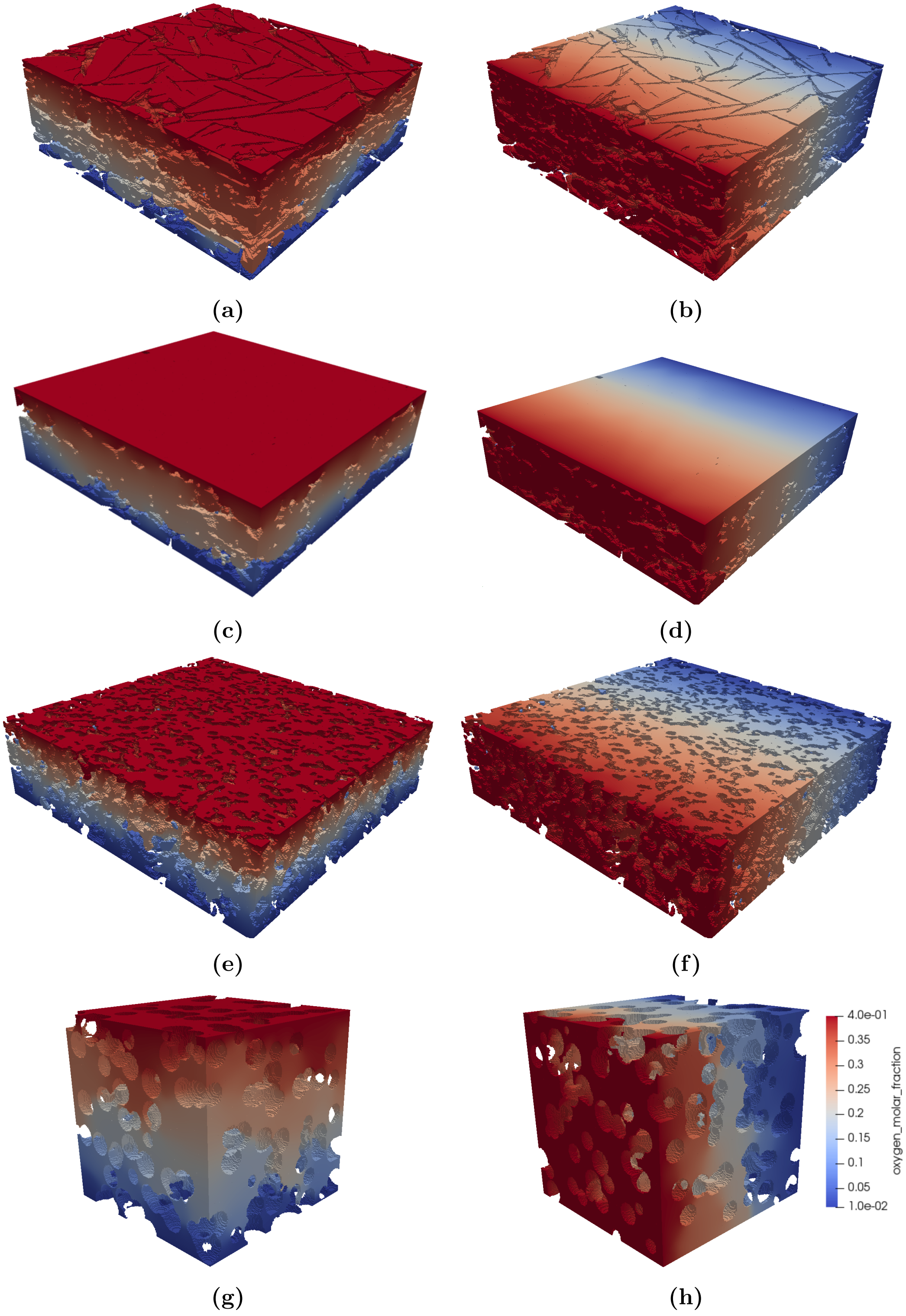
Figures 3.31 and 3.32 show the oxygen molar fraction distributions in the domain for each PTL sample and CL reconstruction obtained from DNS and PNM. The red represents the inlet which has the highest mole fraction, i.e. 0.4, and the blue represents the outlet boundary with a mole fraction of 0.01. The results show a gradual reduction in molar fraction in the corresponding direction, which proves the simulations in both numerical approaches provide physical results.

The formation factor for through-plane and in-plane gas diffusivity of the three PTL samples are shown in Table 3.3. The formation factor is defined as the ratio of the effective diffusivity by the bulk gas diffusivity. Since the transport simulations can be performed using either equivalent diameter or inscribed diameter in PNM, the results from both diameter cases are included. Experimental results for the Toray 120C are obtained from literature [31, 79, 117–120] and those for the other materials were measured experimentally in our laboratory. As shown in the table, the PNM formation factors obtained from the inscribed diameter are significantly lower than the experimental value. The results obtained from the equivalent diameter appear, therefore, to be more reasonable.

For the Toray 120C, the estimated formation factor for DNS simulations in the in-plane and through-plane directions are 0.442 and 0.223, respectively. The formation factors in both directions are in good agreement with the experimental ranges of 0.31-0.54 and 0.14-0.33, respectively. The formation factors computed by the PNM equivalent diameter are 0.414 and 0.167 in the in-plane and through-plane directions, respectively. Although the results from PNM are lower than those from DNS, they are also in agreement with the range of experimental data from literature.

In the case of 39BA, the average formation factors from DNS are 0.415 and 0.610 in the through-plane and in-plane directions, respectively. These are in very good agreement with the experimental values of 0.414 and 0.602. The formation factors from PNM, on the other hand, are 0.283 and 0.492 in each direction. It can be seen that the formation factors from PNM are nearly 26% and 31% lower in the in-plane and through-plane directions, respectively. Though the domain size used for the 39BA simulation was 313 and 1074 μm in through-plane and in-plane directions respectively, the number of pores generated from the images were around 2200 due to its high porosity and large pore regions. The number of pores generated from the Toray 120C was 5300. Considering the number of pores used in the Toray 120C case, the number of pores used for 39BA simulations in PNM might not be sufficient to predict the formation faction in 39BA.

For Ti, the through-plane formation factor obtained from DNS, 0.259, is within the experimental range. The value obtained from PNM, 0.229, is lower than the DNS results, but also within the experimental range. The results from both approaches show around ±6% discrepancies from the experimental result. In the in-plane direction, the average formation factors are 0.177 for DNS and 0.159 for PNM. The



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.31 – Oxygen molar fraction profiles from DNS for a) Toray 120C in the through-plane, and b) in-plane directions, c) SGL 39BA in the through-plane, and d) in-plane directions, e) Ti in the through-plane, and f) in-plane directions, g) CL in the through-plane, and h) in-plane directions.

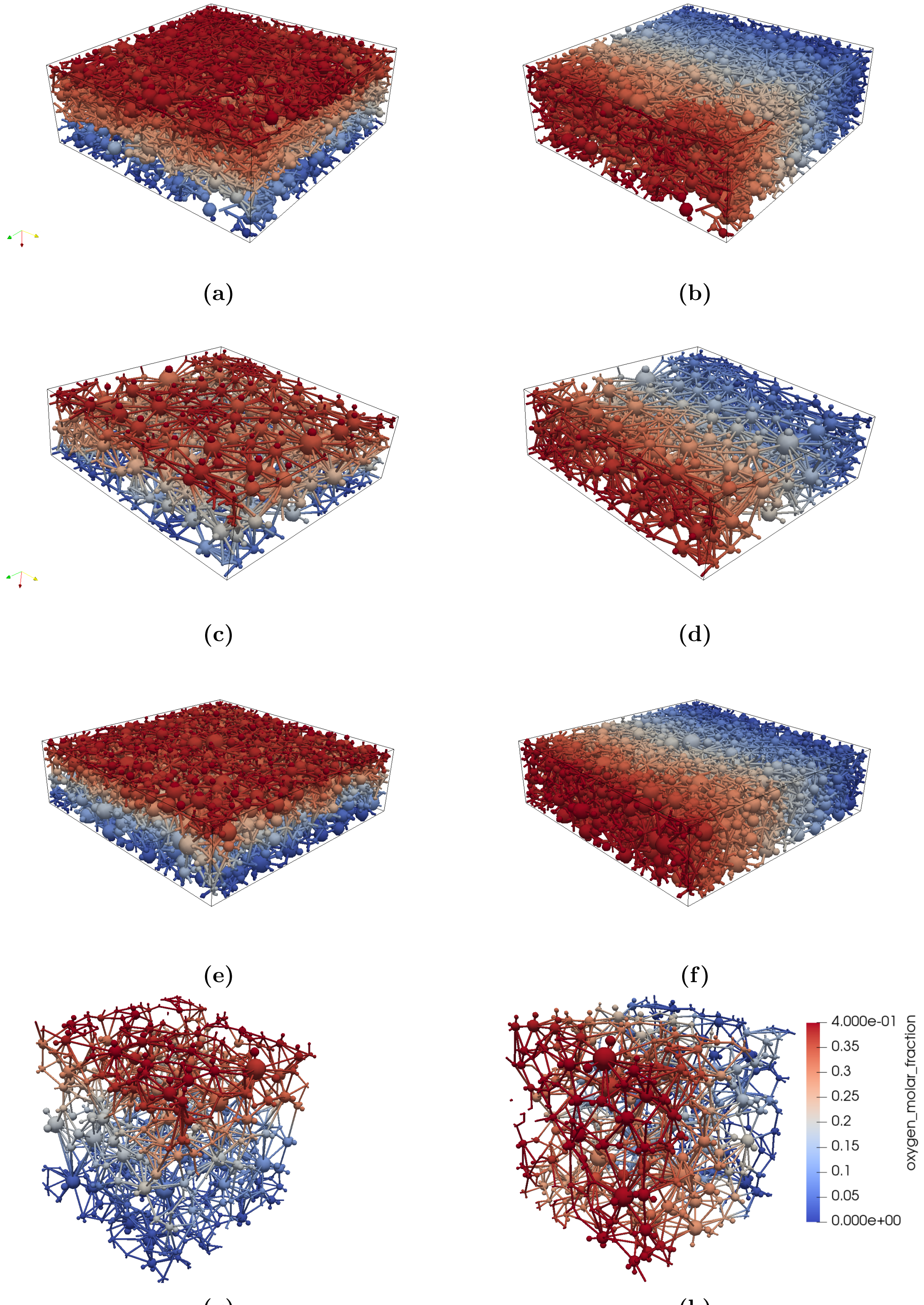


Figure 3.32 – Oxygen molar fraction profiles from PNM for a) Toray 120C in the through-plane, and b) in-plane directions, c) SGL 39BA in the through-plane, and d) in-plane directions, e) Ti in the through-plane, and f) in-plane directions, g) CL in the through-plane, and h) in-plane directions.

Table 3.3 – The dry effective diffusivity and permeability of the PTLs and CL reconstructions compared to the experimental results.

		Direction	DNS	PNM		Experimental
				Inscribed	Equivalent	
Toray 120C	D_{eff}/D_{bulk}	Z	0.223 ± 0.017	0.085 ± 0.009	0.168 ± 0.013	0.14-0.33 [79, 118–120]
		X/Y	0.431 ± 0.032 / 0.452 ± 0.014	0.210 ± 0.011 / 0.206 ± 0.009	0.392 ± 0.014 / 0.387 ± 0.012	0.31-0.54 [80, 120]
	K [m^2]	Z	0.336^*	0.079 ± 0.01	0.405 ± 0.065	0.5-1.13 [79, 118, 121, 122]
	($\times10^{11}$)	X/Y	0.925^* / 0.946^*	0.210 ± 0.013 / 0.199 ± 0.008	0.925 ± 0.053 / 0.882 ± 0.054	0.90-1.83 [79, 81, 82, 123]
SGL 39BA	D_{eff}/D_{bulk}	Z	0.415 ± 0.016	0.149 ± 0.010	0.283 ± 0.008	0.414 ± 0.04
		X/Y	0.602 ± 0.01 / 0.619 ± 0.01	0.304 ± 0.014 / 0.353 ± 0.021	0.445 ± 0.041 / 0.506 ± 0.046	0.602 ± 0.005
	K [m^2]	Z	4.245^*	0.436 ± 0.023	4.960 ± 0.925	3.73 ± 0.12
	($\times10^{11}$)	X/Y	7.083^* / 7.010^*	1.261 ± 0.177 / 1.550 ± 0.167	4.975 ± 0.437 / 6.406 ± 0.916	5.78 ± 0.038
Ti	D_{eff}/D_{bulk}	Z	0.259 ± 0.005	0.116 ± 0.003	0.231 ± 0.009	0.20-0.35
		X/Y	0.188 ± 0.002 / 0.166 ± 0.007	0.100 ± 0.0004 / 0.090 ± 0.011	$0.177 \pm 0.0.001$ / 0.140 ± 0.002	0.27 ± 0.008
	K [m^2]	Z	0.469 ± 0.031	0.114 ± 0.008	0.566 ± 0.062	0.36-0.54
	($\times10^{11}$)	X/Y	0.334^* / 0.278^*	0.100 ± 0.002 / 0.074 ± 0.004	0.348 ± 0.022 / 0.253 ± 0.019	0.63 ± 0.027
CL	D_{eff}/D_{bulk}	Z	0.027 ± 0.002	3.21×10^{-9}	0.029 ± 0.002	0.028 ± 0.03 [81]
		X/Y	0.026 ± 0.002 / 0.027 ± 0.003	3.44×10^{-9} / 2.90×10^{-9}	0.028 ± 0.001 / 0.029 ± 0.001	
	K [m^2]	Z	$6.02\times10^{-2} \pm 9.33\times10^{-4}$	1.61×10^{-2}	$7.71\times10^{-2} \pm 9.06\times10^{-3}$	4.1 ± 0.5 [81]
	($\times10^{16}$)	X/Y	$0.214 \pm 2.0\times10^{-2}$ / $0.147 \pm 6.19\times10^{-3}$	1.73×10^{-2} / 1.46×10^{-2}	$7.38\times10^{-2} \pm 6.05\times10^{-2}$ / $7.94\times10^{-2} \pm 4.51\times10^{-2}$	

* Permeability is obtained from a mesh coarsened twice in all directions (400x400xδ_t is coarsened to 200x200x$\frac{\delta_t}{2}$)

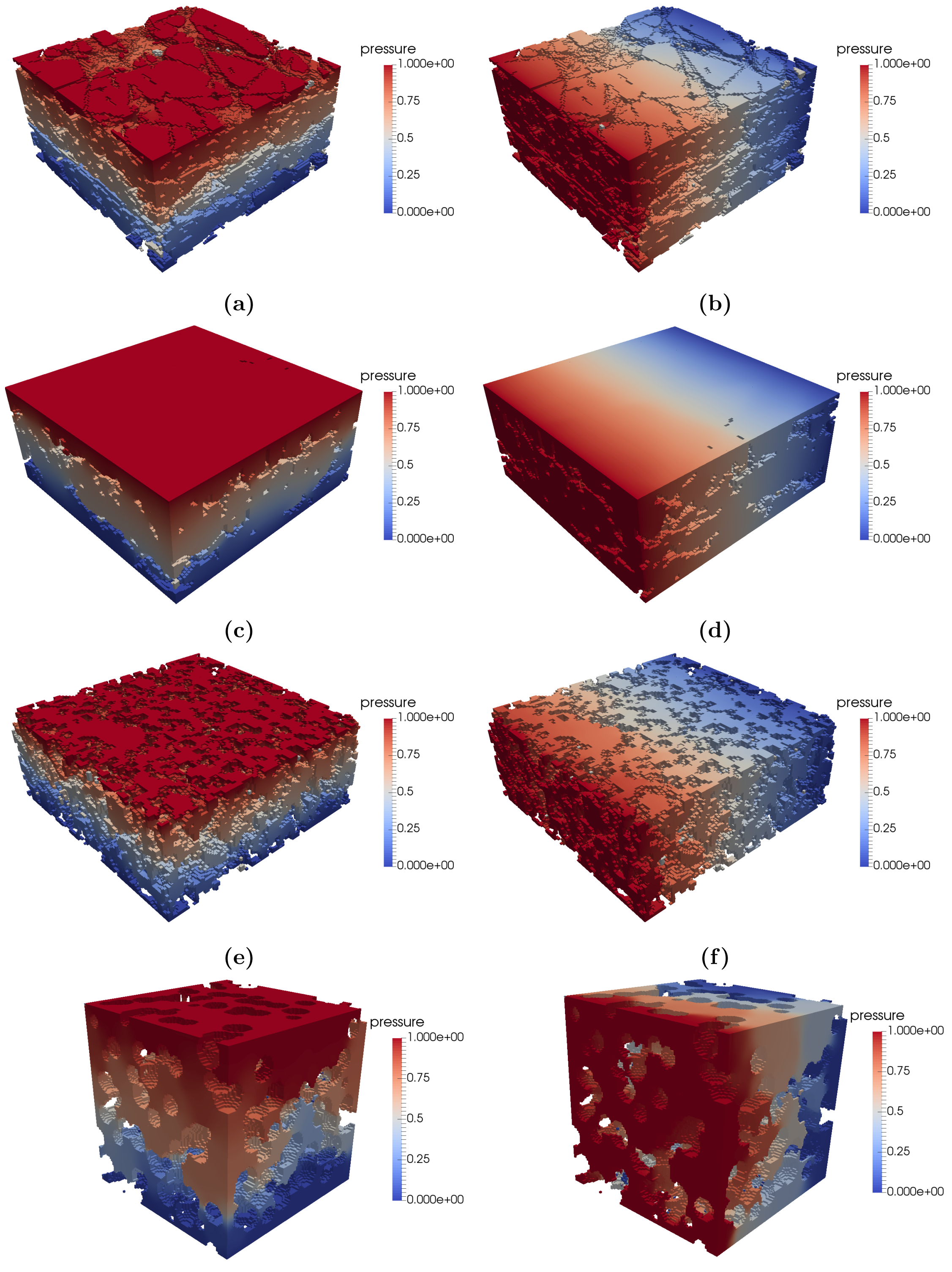
experimental in-plane formation factor is 0.27, which is 50% and 70% greater than the DNS and PNM results. The results from both simulation methods show that the through-plane formation factor is around 31% higher than the in-plane result. However, this trend is not shown in the experimental case and the in-plane result is 10% higher than the through-plane result. This could be attributed to the air leak during the in-plane diffusion experiments. The diffusion bridge designed for the in-plane gas transport requires at least 1 × 5 cm^2 of material size. However, the Ti sample used for the experiments was only 1 × 2 cm^2 due to limited material availability. The rubber gaskets were placed on the rest of the bridge to prevent the gas leak. Because of this reason, there could be unintended gas leak which will increase the gas diffusivity of this case.

For the CL stochastic reconstructions, both DNS and PNM predicted a nearly identical formation factor compared to the experimental result. The experimentally measured formation factors of a CL sample with Pt loading of 0.125 mg/cm^2 and Nafion loading of 30% were obtained by Xu [81] in our laboratory. As discussed in Section 3.1.2, the CL stochastic reconstructions were generated to represent the microstructural characteristics and transport properties of the FIBSEM reconstructions of a CL sample with a Pt loading of 0.025 mg/cm^2 and ionomer loading of 30% by weight. Since the same simulation results were already published by Sabharwal et al. [30], this work aims at validating the result again by comparing the simulations to the experimental data.

3.5 Dry effective permeability

Permeation simulations in PTL and CL domains were performed as discussed in Chapter 2. Figures 3.33 and 3.34 show the pressure distribution for each material in the in-plane and the through-plane directions obtained from DNS and PNM. The red face represents the inlet face with the highest pressure and the blue face represents the outlet boundary. As can be seen in the figures, all cases show gradual decrements in pressure within their domains, which prove the simulations in PNM and DNS returned physical results. The elapsed computational time for PNM is the same as mentioned in Section 3.4 because all diffusion and permeation simulations are performed at once. For the DNS simulations, the meshes were obtained from coarsened images as discussed in Section 2.3.2.1. Each simulation took 4–8 hours using 15 cores with about 30–50 million DOFs. The computations were enabled in part by support provided by WestGrid [124] and Compute Canada [125].

The dry effective permeability of the three PTLs in the through-plane and in-



(a) **(b)**

(c) **(d)**

(e) **(f)**

(g) **(h)**

Figure 3.33 – Pressure profiles from DNS for a) Toray 120C in the through-plane, and b) in-plane directions, c) SGL 39BA in the through-plane, d) in-plane directions, e) Ti in the through-plane, and f) in-plane directions, and g) CL reconstruction in the through-plane and h) in-plane directions.

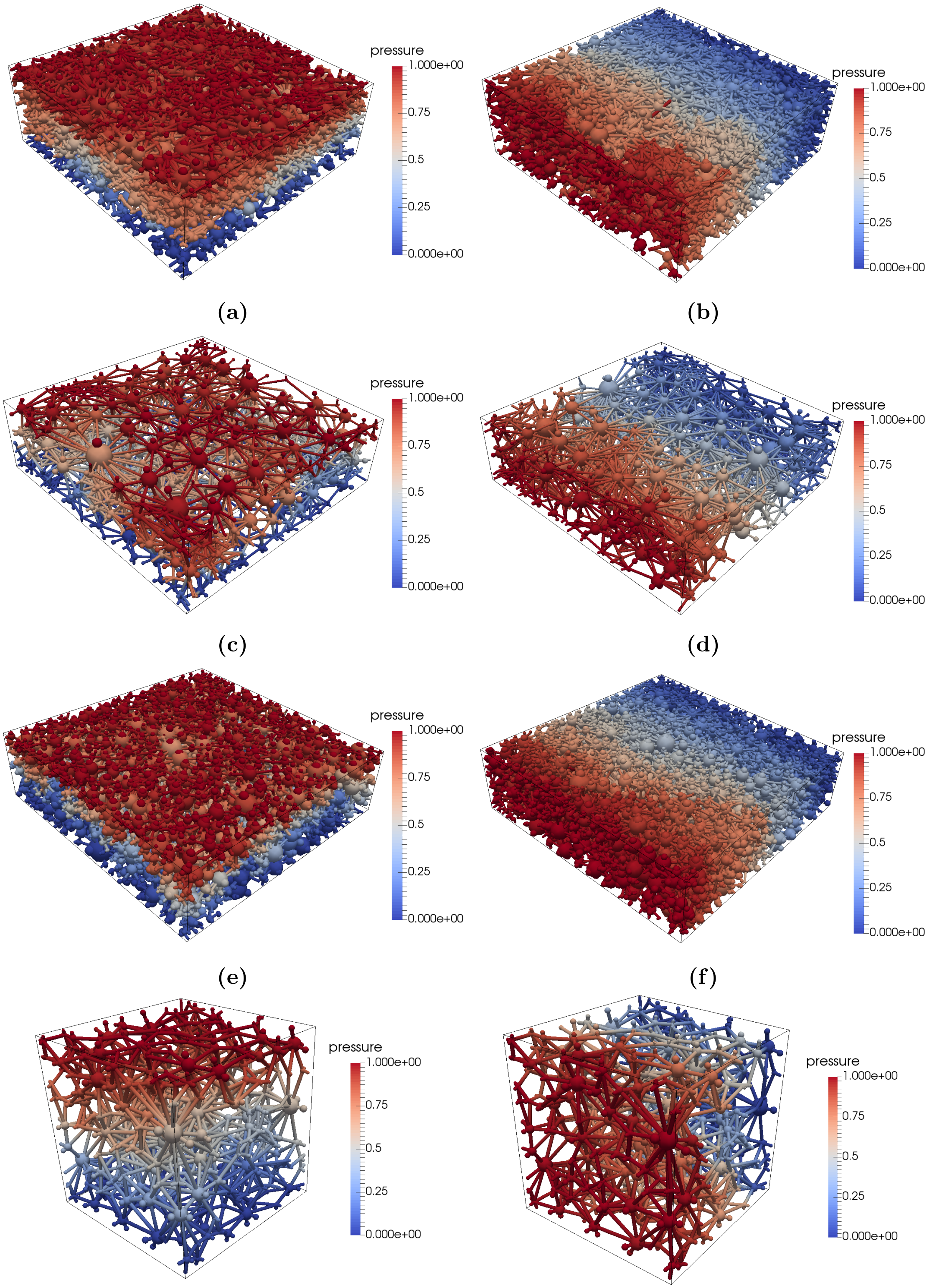
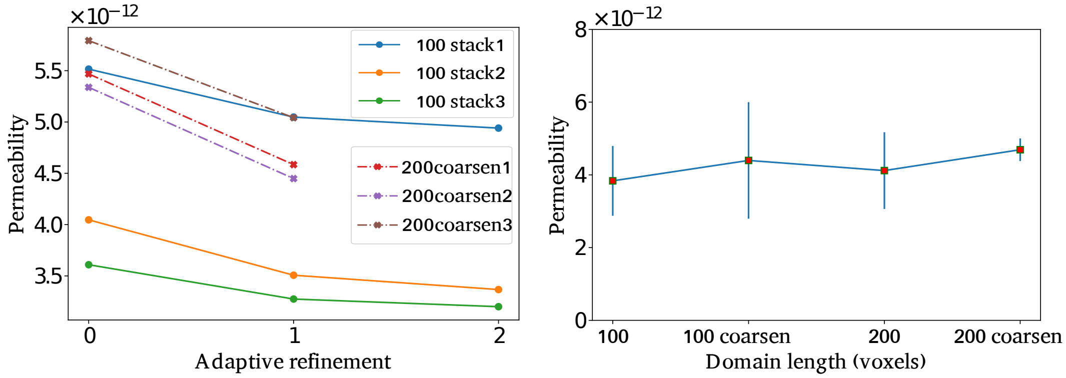


Figure 3.34 – Pressure profiles from DNS for a) Toray 120C in the through-plane, and b) in-plane directions, c) SGL 39BA in the through-plane, d) in-plane directions, e) Ti in the through-plane, and f) in-plane directions, and g) CL reconstruction in the through-plane and h) in-plane directions.



(a) **(b)**

Figure 3.35 – a) Mesh accuracy tests using different levels of adaptive mesh refinement, and b) domain size analysis using the Ti meshes. The domain lengths 100 and 200 represent the number of voxels in the in-plane directions. The coarsened cases are obtained from larger images by coarsening twice in the in-plane directions, e.g., images with 400 voxels in one direction are coarsened to 200 voxels.

plane directions are also shown in Table 3.3. The permeability results from PNM show again that using the inscribed diameter is not appropriate. The values obtained from the equivalent diameter, however, are comparable with the experimental results.

The effective permeability in DNS is computed by solving the Stokes flow equations using the incompressible fluid flow module in OpenFCST. In order to obtain reliable simulation results from a mesh, adaptive refinement is applied to refine the mesh elements near the solid wall. Figure 3.35a shows the permeability results according to different levels of adaptive refinement. For the meshes with the in-plane length of 100 voxels, the permeability results are decreased and stabilized at the refinement level of 2. This is because the refined mesh elements near solid walls are better able to account for boundary layers of the fluid. Based on the results from three of the 100 length meshes, level 1 adaptive refinement returns sufficiently reliable values with moderate computation cost. Due to this reason, the 200 in-plane length meshes, which are obtained by coarsening 400 length images, are tested with 1 level of adaptive refinement, and returned large decreases compared to non-refined meshes. The reason for the steeper slopes in the coarsened meshes than the original meshes is because of a larger voxel size of coarsened images.

In order to determine the representative size of the coarsened meshes, three different sizes were selected, and three meshes for each domain size case were generated. Figure 3.35b shows the through-plane permeability variations for each mesh case. Mesh elements near solid walls were refined adaptive based on the mesh size for better accuracy, i.e., twice for the 100 size meshes, and once for the 200 size meshes. The original voxel meshes, i.e. 100 and 200 in Figure 3.35b, show ±10% of uncertainty, which explains these domains are not sufficiently large to represent the permeability. The coarsened 100 results in Figure 3.35 shows a larger mean value than the 200 cases. This explains that coarsening in the in-plane directions increases the through-plane permeability. The 100 coarsened case shows a larger uncertainty than the 200 cases. The 200 coarsened case (obtained from 400×400×160 stacks) shows the smallest uncertainty (±3%), which implies the 400×400×160 domain can sufficiently represent the permeability of the material.

The tests for the mesh refinement level and the domain size, however, were tested only for the Ti mesh in the z-direction due to limited time and computational resources. For all the other results, i.e., Ti in the x- and y-direction; Toray 120C and SGL 39BA in all directions, smaller meshes were generated, and one simulation was performed for each direction. Instead of keeping the full thickness, the original images were coarsened in all directions ($400\times400\times\ \delta_t$ is coarsened to $200\times200\times\ \frac{\delta_t}{2}$). For this reason, the reliability of these results is lower than the results of the Ti z-direction. The domain size and mesh refinement tests should be done for all cases in future work.

For the Toray 120C, the results from DNS and PNM in both directions are lower than the experimental results. The permeability results from DNS are 0.336×10^{-11} m^2 in the through-plane direction and the mean in-plane permeability is 0.936×10^{-11} m^2. The PNM through-plane permeability (0.408×10^{-11} m^2) is 22% lower than the lower reference value of 0.5×10^{-11} m^2, and the average in-plane permeability (0.903×10^{-11} m^2) is similar to the in-plane lower reference value of 0.90×10^{-11} m^2. Permeability simulations using the same Toray 120C images were reported by Garcia-Salaberri et al. [49]. They employed a larger domain with size $0.78\times0.78\times0.367$ mm^3 and the permeability results were obtained using lattice Boltzmann method (LBM) in through-plane and in-plane directions, resulting in 0.55×10^{-11} and 1.7×10^{-11} m^2, respectively. Moosavi et al. [33] reported the permeability of Toray 120C using the finite volume method, and their results for the through-plane gas permeability was 0.579×10^{-11} m^2. The number of through-plane voxels in their simulations were 280. This means that they employed more surface pores than the meshes for Toray 120C in this work. As the surface pores have larger pore sizes, reducing the thickness would decrease the mass flux through the domain, which results in a lower permeability. Due to this reason, the lower permeability than the references obtained for Toray 120C in this work is reasonable.

The 39BA PTL through-plane results from DNS and PNM are 4.245×10^{-11} m^2

and 4.960×10^{-11} m^2, respectively. The average in-plane results are 7.046×10^{-11} m^2 from DNS and 5.690×10^{-11} m^2 from PNM. The in-plane and through-plane permeabilities from DNS are 13% and 20% higher than the experimental values, respectively. The through-plane permeability from PNM is 33% higher than the experimental results. The average in-plane permeability from PNM shows a small discrepancy, however, the x-direction permeability is nearly identical to the through-plane permeability. This is unreasonable due to the anisotropy of the material. All the simulation results show higher predictions than the experimental values for the 39BA material. There could be several reasons for the higher simulation results. First, missing small pores in the carbon matrix would lower the flow resistance in the domain, which would result in a higher mass flux. Second, when the material is placed in the diffusion bridge for the experiments, a little compression of the material is unavoidable. The effect of compression reduces effective transport in the material [26]. Due to these reasons, it is reasonable that the transport simulations with uncompressed materials show higher permeabilities.

In the case of Ti, the through-plane permeability from PNM is 0.566×10^{-11} m^2, which shows is 4% larger than the experimental range. The through-plane results from DNS are 0.469×10^{-11} m^2. In the in-plane direction, the average results are 0.300×10^{-11} m^2 and 0.306×10^{-11} m^2 from PNM and DNS, respectively. The simulation results show a larger through-plane permeability than the in-plane values, which is in agreement with the anisotropy observed in the diffusion case. The experimentally measured in-plane permeability is about 0.63×10^{-11} m^2, which is 17% greater than the through-plane result. As discussed in the dry diffusivity section, the in-plane experimental results could be inaccurate due to a gas leak in the diffusion bridge, since the permeation experiments uses the same diffusion bridge as for the diffusion experiments.

For the CL stochastic reconstructions, the simulation results from both DNS and PNM are order of 10^{-18} m^2, but the experimental results reported by Xu [81] were around 4.1 ×10^{-16} m^2. In their study, the effective pore diameter is 7.0 nm, which will result in significant Kundsen slip effects. The Kundsen effect is considered in the diffusion simulations for both PNM and DNS, whereas it is not implemented for the permeation in both methods. As a result, the permeability results from both PNM and DNS are lower than the experimental values. The Knudsen effect should be incorporated for the permeation simulations in future work.

In general, both DNS and PNM predicted the dry permeability of each material reasonably. The DNS results were more reliable even though the simulations were done with the coarsened meshes. PNM could not show the anisotropy in 39BA properly, however, the results were generally similar to those from DNS.

3.6 Relative diffusivity

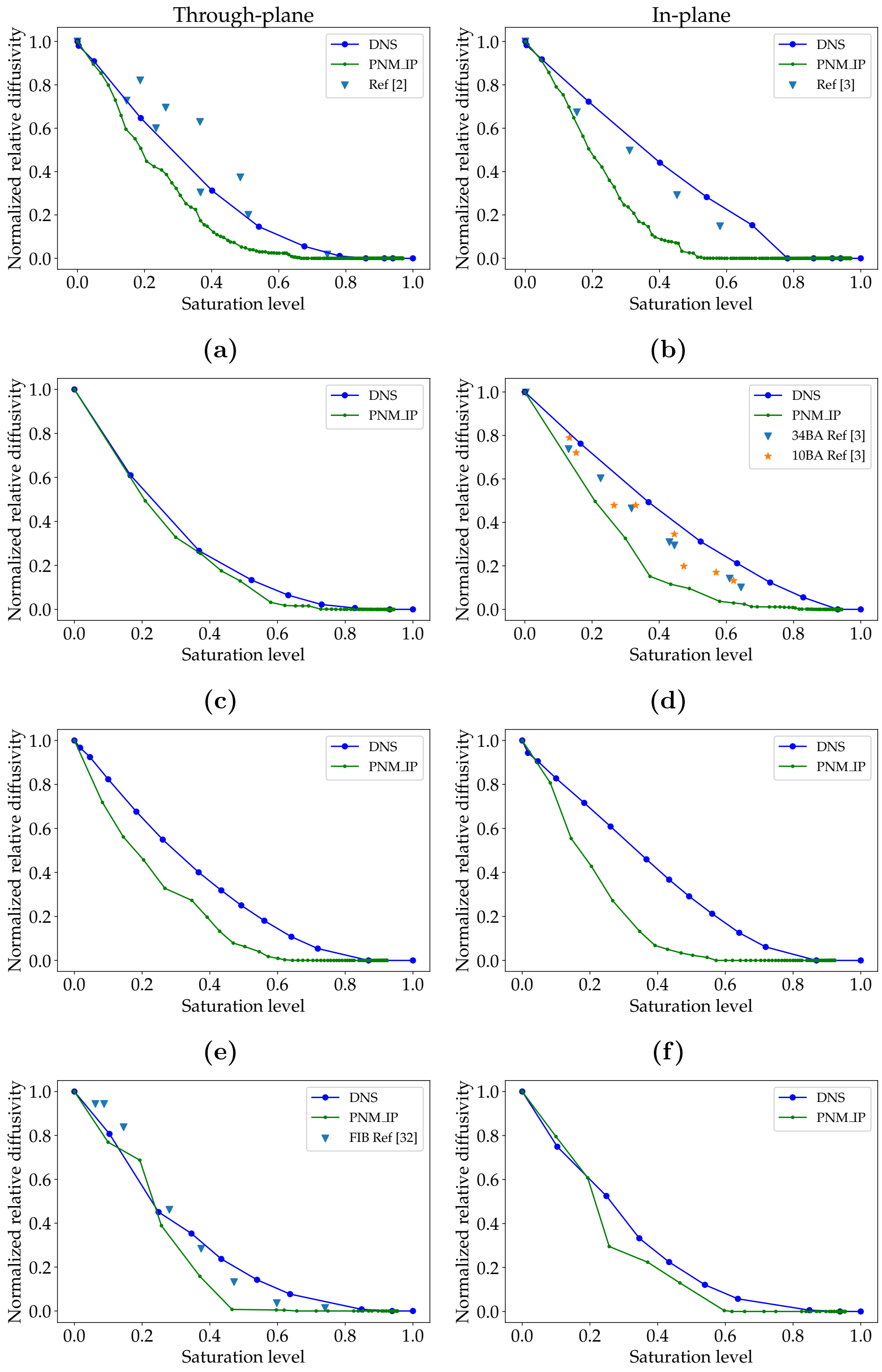
The formation factor at varying saturations are computed using both PNM and DNS. PNM makes an array of intruded pores, and exclude them from the transport simulations as discussed in Section 2.2.6. In DNS, the non-wetting phase is intruded in the domain using the FM algorithm as mentioned in Section 2.3.3 and then the computational mesh is generated only of the not intruded pore voxels. The main difference between PNM and DNS in terms of the transport simulation under saturated condition is that the DNS method needs separate image stacks at each saturation from the FM algorithm.

Figure 3.36 shows the calculated relative formation factors of the three PTLs in the in-plane and through-plane directions. The relative diffusivity is normalized by the dry diffusivity results since dry diffusivities are different for DNS and PNM. The normalization is used to better compare. Even though PNM provides two percolation algorithms (i.e. invasion percolation (IP) and ordinary percolation (OP)), the relative diffusivity were computed only for the IP case, since IP and OP return the same intrusion results.

Figures 3.36a and 3.36b show the relative diffusivity for Toray 120C. Hwang and Weber [2] reported the through-plane relative diffusivity for Toray 120C measured experimentally as shown in Figure 3.36a. Likewise, Tranter et al. [3] measured the in-plane relative diffusivity of Toray 120 with 5% PTFE content. In both directions, the results from DNS show better prediction than PNM. The relative diffusivity results from PNM are lower than the DNS result at saturations between 0.1–0.7. At the saturation level of 0.7, PNM results are zero. In the in-plane direction, DNS and PNM show a larger discrepancy to each other. PNM predicts lower values than DNS through the entire saturation range.

The results for 39BA are shown in Figures 3.36c and 3.36d. The in-plane results are compared with experimental results of SGL 34BA and SGL 10BA measured by Tranter et al. [3] shown in Figure 3.36d. Both reference materials have 5% PTFE content, and the thickness of 10BA and 34BA are 400 μm and 280 μm, respectively. In the through-plane direction, both DNS and PNM show very similar results to each other. The DNS results show a much closer prediction than the PNM results in the in-plane direction.

The reason for the noted faster change in PNM is related to the water injection from the bottom side. As the porosity of the surface region is higher than the internal



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.36 – Variation of the formation factors in the through-plane and in-plane directions with saturation: a) and b) Toray 120C in through-plane and in-plane directions compared to previously reported literature data for Toray 120 GDLs [2, 3], c) and d) SGL 39BA in through-plane and in-plane directions, e) and f) Ti in through-plane and in-plane directions, and g) and h) for CL reconstructions. The in-plane results for 39BA are compared to results of SGL 34BA and SGL 10BA from the literature [3].

region for fibrous material, the surface region of the 39BA would be mostly large pores. Injecting water from the bottom side gradually occupies these large pores in the bottom surface. Then, the gas flux would decrease at the inlet face since the number of pores through which the gas can pass decreases. At some point when most of the large pores at the bottom face are saturated, the gas flux would decrease significantly because gas flow can pass through only small pores. PNM, however, uses less pore size data than FM, which would result in a faster blockage of the inlet face of the non-wetting phase. The same trends are shown in the Toray 120C case in Figure 3.36a.

The relative diffusivity curves for Ti are shown in Figure 3.36e and Figure 3.36f. It can be seen that the DNS results are larger than the PNM results through all saturations, but the discrepancy is not significant. PNM shows a faster decreasing function that reaches 0 at the saturation level of 0.6. DNS shows some amount of mass flux until 0.7 saturation. In the in-plane direction, the PNM results show larger discrepancies than the through-plane results compared to DNS. As shown in the figure, the in-plane diffusivity of PNM decreases fast in the saturation range of 0–0.4. This implies that the pores intruded by the non-wetting phase more penalize the gas flow than the DNS case.

The relative diffusivity of the CL reconstructions are shown in Figures 3.36g and 3.36h. As discussed in the water intrusion section, the liquid water injection in the CL occurs within the domain, and the CL stochastic reconstructions do not have anisotropy, which means the relative diffusivity curves are supposed to be the same in the in-plane and through-plane directions. The curves from DNS show nearly identical and smooth decreasing functions whereas the PNM results show more fluctuations in both directions. The simulation results are compared to the relative diffusivity computed using the FIBSEM images reported by Sabharwal et al. [29]. The DNS result is in good agreement with the reference and the PNM results in Figure 3.36h also follows the DNS results similarly.

Chapter 4

Conclusions and future work

The objective of this thesis was:

- Create tomography images of morphologically different porous media using μCT.
- Simulate liquid water and mercury intrusion in porous media to compare simulation results from PNM and DNS.
- Use tomography images to predict dry permeability and diffusivity of the porous media with the two numerical approaches and evaluate them by comparing to experimental properties.
- Estimate relative transport properties and compare the results between the PNM and DNS approaches.

In this thesis, two different numerical approaches were used to estimate characteristics of microstructure and transport properties of commercial porous transport layers (PTLs). The two numerical approaches studied were pore network modeling (PNM) and continuum based direct numerical simulation (DNS) methods. In both cases, μCT images were employed to extract a pore network and generate a mesh to perform simulations.

Two fibrous GDL and one PTL samples (Toray 120C, SGL 39BA, and Ti sinter) were scanned using a μCT to obtain tomographic images. Since the fibrous materials (Toray 120C and SGL 39BA) were based on carbon fibres, the x-ray voltage was set to 25–30 kV whereas it was 90–100 kV for the Ti powder based sample. A sample was placed vertically on the sample plate using the sample holder. Using a sample holder was essential to obtain proper cross-sectional images since a tilted sample could return cross-sections of an inclined image. After scanning the samples, the raw tomography images were reconstructed and then the reconstructed images were cropped into a

$1500 \times 1500 \times$ thickness, δ_t. The image stack for each sample was then used as an original stack to be binarized.

Image segmentation was performed on the original image stacks. Obtaining accurate binary images was crucial since small errors in the segmentation could lead to erroneous results. For an accurate segmentation, a given raw image must be of high quality and contain very little noise. The bilateral filter was applied to the original images to remove noise in the images. One parameter to be matched after binarization was the porosity. Then, the segmentation quality was visually assessed by overlaying the edges of the binary images onto the filtered images. In this work, the Otsu and Sauvola algorithms were used for the image segmentation. The threshold algorithms that can achieve desired porosities of the samples were selected. The resolved porosity for each sample showed a good agreement with experimentally measured values and the overlaid images showed a great segmentation quality. Then, the segmented stacks were cropped into three $600 \times 600 \times \delta_t$ subsamples for further studies.

Statistical functions, such as the two-point correlation function, the chord length function and the pore size distribution, were used to characterize the porous media microstructure. The two-point correlation function and the chord length function were used to show the similarity, or lack thereof, between each PTL subsamples. The pore size distribution were obtained separately from the two different numerical methods (PNM and DNS). PNM extracts a network using the watershed segmentation algorithm implemented in PoreSpy [43]. The pore network contains geometric information, such as pore and throat sizes, which can be used to obtain pore and throat size distributions. In PNM, using the inscribed diameters was better able to predict reasonable pore sizes than the equivalent diameters. In DNS, the sphere fitting algorithm was used to compute pore sizes by assigning pore radii to all voxels. PSDs obtained from PNM could not show the pore size data of the materials precisely.

To compare the predicted PSDs, virtual MIP was performed for both approaches. The mercury intrusion curves from both cases returned nearly identical results, and showed a good agreement to the experimental MIP curves. From the predicted MIP curves, MIP-PSDs were computed by taking derivative with respect to pore sizes. The MIP-PSDs and the experimental PSDs of the materials were compared to the PSDs computed from the sphere fitting algorithm and the network extraction. The MIP-PSDs showed nearly identical to the FM_PSDs with only small discrepancies for all cases but CL. The discrepancies were because of the shielding effect, but the discrepancies were not significant. Based on these observations, MIP-PSDs were sufficient to represent the pore sizes of the materials.

If mercury is intruded only using the bottom face boundary, the shielding effect had a larger impact on the intrusion curves than for other materials that using the 6-face boundaries. Larger pores were concealed by smaller pores. This resulted in higher small-sized pore portions than the 6-face boundary cases.

Liquid water intrusion in PTLs showed overall agreement of saturation patterns between FM and PNM, as the pore sizes of the domain were similarly obtained. FM reached the first breakthrough faster than PNM as FM used more pore size data than PNM. The larger number of pore sizes leaded to smoother intrusion trends and more realistic pore filling than PNM. As PNM had less pore connectivity, the shielding effect was larger than in FM, which result in a steeper intrusion than PNM in mid-range pores.

Dry transport properties for the three PTLs were computed using the PNM and DNS methods. For the PNM cases, since it provides two pore sizes, i.e. inscribed diameter and equivalent diameter, transport properties from both diameter cases were computed. The results using the equivalent diameters were in better agreement with the experimental values. The inscribed diameter results were lower than the reference data. The results from the DNS were in great agreement with the experimental results in all cases. The dry diffusivity results from PNM were good for the Toray 120C and Ti cases, but they were too low for 39BA. For permeability, not all results from PNM matched well to the experimental results. The permeability simulations in DNS were also performed, and showed a reasonable agreement with the reference data. Therefore, DNS is more accurate and more reliable than PNM. PNM should use the equivalent diameters to estimate transport properties.

To compute the relative transport properties, water intrusion simulations were performed. The relative transport properties from PNM were computed by restricting the pores that were intruded. DNS showed a better agreement to available literature data than the PNM results for the relative diffusivity. PNM result for the throgh-plane 39BA showed a similar relative diffusivity to the DNS results. Other than this, PNM results showed lower relative diffusivity than the DNS results in all cases because the liquid water clusters in PNM blocked the gas outlet face faster than in the FM algorithm.

In general, both DNS and PNM are very useful techniques for a pore scale analysis. DNS results are more accurate and reliable than PNM. However, PNM is faster as it sacrifices some information from the source images. In PNM, in order to compute PSDs, the inscribed diameter method is recommended, but when it comes to computing transport properties, using the equivalent diameter gives more reasonable results. DNS uses all image voxels for simulations, which results in a great accuracy but high computational cost.

4.1 Future work

The observations in this work have showed the possibilities of both PNM and DNS methods for further investigations of 3D microstructure. Both numerical approaches showed very close predictions to experimental results when it comes to pore size distribution (PSD) and mercury intrusion porosimetry (MIP). The results for the liquid water intrusion, however, showed discrepancies from experimental references. For the liquid water intrusion, only one contact angle was used by assuming all the internal materials were one solid. An actual GDL, however, contains multiple contents, such as carbon matrix; PTFE; and micro porous layer (MPL). These contents may have critical impacts on the internal water propagation.

The carbon matrix in SGL 39BA was not separated from the solid region. As a result, small pores in the carbon matrix were ignored, which could have made differences of the gas transport in the domain. In this regard, separating the carbon matrix and the carbon fibres would be useful for a better accuracy in transport simulations.

Segmenting the carbon matrix from the SGL fibrous papers with a high accuracy is very difficult due to overlapping environment in grayscale intensities between the carbon matrix, fibres, and pore regions. One method to overcome this limit is using stochastic reconstructions. The GDL generator module in pyFCST is able to generate fibres and the carbon matrix separately. This module can be used to study the optimized GDL and PTL conditions for fuel cell and electrolyzer performances.

References

[1] S. A. Grigoriev, P. Millet, S. A. Volobuev, and V. N. Fateev. Optimization of porous current collectors for PEM water electrolysers. *International Journal of Hydrogen Energy*, 34(11):4968–4973, jun 2009. ISSN 03603199. doi: 10.1016/j.ijhydene.2008.11.056.

[2] G S Hwang and A Z Weber. Effective-Diffusivity Measurement of Partially-Saturated Fuel-Cell Gas-Diffusion Layers. *Journal of The Electrochemical Society*, 159(11):683–692, 2012. doi: 10.1149/2.024211jes. URL `http://jes.ecsdl.org/content/159/11/F683.full.pdf`.

[3] T. G. Tranter, P. Stogornyuk, J. T. Gostick, A. D. Burns, and W. F. Gale. A method for measuring relative in-plane diffusivity of thin and partially saturated porous media: An application to fuel cell gas diffusion layers. *International Journal of Heat and Mass Transfer*, 110:132–141, 2017. ISSN 00179310. doi: 10.1016/j.ijheatmasstransfer.2017.02.096. URL `http://dx.doi.org/10.1016/j.ijheatmasstransfer.2017.02.096`.

[4] National Energy Board. Canada's Energy Transition: Historical and Future Changes to Energy Systems – Update – An Energy Market Assessment - August 2019. 2019. ISSN 2562-5322. URL `https://www.cer-rec.gc.ca/nrg/ntgrtd/mrkt/cndsnrgtrnstn/cndsnrgtrnstn-eng.pdf`.

[5] Environment Canada and Climate Change. Canada's greenhouse gas and air pollutant emissions projections 2018. page 112, 2018. URL `http://publications.gc.ca/collections/collection{_}2018/eccc/En1-78-2018-eng.pdf`.

[6] Jeffrey D Wishart, Zuomin Dong, and Marc M Secanell. OPTIMIZATION OF A PEM FUEL CELL SYSTEM FOR LOW-SPEED HYBRID ELECTRIC VEHICLES. Technical report, 2006. URL `https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-pdf/IDETC-CIE2006/4255X/757/2651558/757{_}1.pdf`.

[7] Jun Shen, Jianqin Zhou, Nelson G.C. Astrath, Titichai Navessin, Zhong Sheng Liu, Chao Lei, Jurandir H. Rohling, Dmitri Bessarabov, Shanna Knights, and Siyu Ye. Measurement of effective gas diffusion coefficients of catalyst layers of PEM fuel cells with a Loschmidt diffusion cell. *Journal of Power Sources*, 196 (2):674–678, jan 2011. ISSN 03787753. doi: 10.1016/j.jpowsour.2010.07.086.

[8] Li Chen, Gang Wu, Edward F. Holby, Piotr Zelenay, Wen Quan Tao, and Qinjun Kang. Lattice boltzmann pore-scale investigation of coupled physical-electrochemical processes in C/PT and non-precious metal cathode catalyst layers in proton exchange membrane fuel cells. *Electrochimica Acta*, 158:175–186, mar 2015. ISSN 00134686. doi: 10.1016/j.electacta.2015.01.121.

[9] Jian Zhao. *Catalyst Layers in Polymer Electrolyte Membrane Fuel Cells: Formation, Characterization and Performance*. Ph.d, University of Waterloo, 2019. URL `https://uwspace.uwaterloo.ca/handle/10012/14425`.

[10] Rüdiger Schweiss, Christian Meiser, Tanja Damjanovic, Ivano Galbiati, and Nico Haak. WHITE PAPER SIGRACET ® Gas Diffusion Layers for PEM Fuel Cells , Electrolyzers and Batteries. Technical report, 2016.

[11] Ryan K. Phillips, Brooks R. Friess, Anthony D. Hicks, Julie Bellerive, and Mina Hoorfar. Ex-situ Measurement of Properties of Gas Diffusion Layers of PEM Fuel Cells. *Energy Procedia*, 29:486–495, jan 2012. ISSN 1876-6102. doi: 10.1016/J.EGYPRO.2012.09.057. URL `https://www.sciencedirect.com/science/article/pii/S1876610212014774`.

[12] Arunkumar Jayakumar, Sundar Pethaiah Sethu, Maximiano Ramos, John Robertson, and Ahmed Al-Jumaily. A technical review on gas diffusion, mechanism and medium of PEM fuel cell. doi: 10.1007/s11581-014-1322-x.

[13] Seoung-Ju Lee, Jung Hun Yoo, Bo Shim, and Sung-Cul Yi. Ceramic Processing Research Microstructural analysis of mass transport phenomena in a PEM fuel cell cathode. 17(7):773–777, 2016. URL `http://jcpr.kbs-lab.co.kr/file/JCPR{_}vol.17{_}2016/JCPR17-7/23.2016-125{_}773-777.pdf`.

[14] J. P. Feser, A. K. Prasad, and S. G. Advani. Experimental characterization of in-plane permeability of gas diffusion layers. *Journal of Power Sources*, 162(2 SPEC. ISS.):1226–1231, nov 2006. ISSN 03787753. doi: 10.1016/j.jpowsour.2006.07.058.

[15] G. Bruggeman. Calculation of various physics constants in heterogeneous substances i dielectricity constants and conductivity of mixed bodies from isotropic substances. *Annalen der Physik*, 416(7):pp. 636–664., 1935. doi: 10.1002/andp.19354160705.

[16] Soowhan Kim and M M Mench. Investigation of Temperature-Driven Water Transport in Polymer Electrolyte Fuel Cell: Phase-Change-Induced Flow. *Journal of The Electrochemical Society*, 156:353–362, 2009. doi: 10.1149/1.3046136. URL `http://jes.ecsdl.org/content/156/3/B353.full.pdf`.

[17] Zhong Xie, Titichai Navessin, Ken Shi, Robert Chow, Qianpu Wang, Datong Song, Bernhard Andreaus, Michael Eikerling, Zhongsheng Liu, and Steven Holdcroft. Functionally Graded Cathode Catalyst Layers for Polymer Electrolyte Fuel Cells II. Experimental Study of the Effect of Nafion Distribution. 2005. doi: 10.1149/1.1904990.

[18] Jin Hyun Nam, Kyu Jin Lee, Gi Suk Hwang, Charn Jung Kim, and Massoud Kaviany. Microporous layer for water morphology control in PEMFC. *International Journal of Heat and Mass Transfer*, 52(11-12):2779–2791, may 2009. ISSN 00179310. doi: 10.1016/j.ijheatmasstransfer.2009.01.002.

[19] Gu Gon Park, Young Jun Sohn, Tae Hyun Yang, Young Gi Yoon, Won Yong Lee, and Chang Soo Kim. Effect of PTFE contents in the gas diffusion media on the performance of PEMFC. *Journal of Power Sources*, 131(1-2):182–187, may 2004. ISSN 03787753. doi: 10.1016/j.jpowsour.2003.12.037.

[20] Rangachary Mukundan, John R Davey, Joseph D Fairweather, Dusan Spernjak, Jacob Spendelow, Daniel S Hussey, David L Jacobson, Peter Wilde, Ruediger Schweiss, and Rod L Borup. Effect of Hydrophilic Treatment of Microporous Layer on Fuel Cell Performance. doi: 10.1149/1.3484604.

[21] R. Schweiss, M. Steeb, and P. M. Wilde. Mitigation of Water Management in PEM Fuel Cell Cathodes by Hydrophilic Wicking Microporous Layers. *Fuel Cells*, 10(6):1176–1180, dec 2010. ISSN 16156846. doi: 10.1002/fuce.201000003. URL `http://doi.wiley.com/10.1002/fuce.201000003`.

[22] M. P. Manahan, M. C. Hatzell, E. C. Kumbur, and M. M. Mench. Laser perforated fuel cell diffusion media. Part I: Related changes in performance and water content. *Journal of Power Sources*, 196(13):5573–5582, jul 2011. ISSN 03787753. doi: 10.1016/j.jpowsour.2011.01.014.

[23] D. Gerteisen, T. Heilmann, and C. Ziegler. Enhancing liquid water transport by laser perforation of a GDL in a PEM fuel cell. *Journal of Power Sources*, 177 (2):348–354, mar 2008. ISSN 03787753. doi: 10.1016/j.jpowsour.2007.11.080.

[24] R. Alink, J. Haußmann, H. Markötter, M. Schwager, I. Manke, and D. Gerteisen. The influence of porous transport layer modifications on the water management in polymer electrolyte membrane fuel cells. *Journal of Power Sources*, 233:358–368, 2013. ISSN 03787753. doi: 10.1016/j.jpowsour.2013.01.085.

[25] Jeff T Gostick, Marios A Ioannidis, Michael W Fowler, and Mark D Pritzker. Pore network modeling of fibrous gas diffusion layers for polymer electrolyte membrane fuel cells. *Journal of Power Sources*, 173(1):277–290, 2007. ISSN 0378-7753. doi: https://doi.org/10.1016/j.jpowsour.2007.04.059. URL `http://www.sciencedirect.com/science/article/pii/S0378775307009056`.

[26] T. G. Tranter, J. T. Gostick, A. D. Burns, and W. F. Gale. Pore Network Modeling of Compressed Fuel Cell Components with OpenPNM. *Fuel Cells*, 16 (4):504–515, 2016. ISSN 16156854. doi: 10.1002/fuce.201500168.

[27] Mohammadreza Fazeli, James Hinebaugh, Zachary Fishman, Christian Tötzke, Werner Lehnert, Ingo Manke, and Aimy Bazylak. Pore network modeling to explore the effects of compression on multiphase transport in polymer electrolyte membrane fuel cell gas diffusion layers. *Journal of Power Sources*, 335:162–171, 2016. ISSN 03787753. doi: 10.1016/j.jpowsour.2016.10.039.

[28] J.K. Lee, CH. Lee, and A. Bazylak. Pore network modelling to enhance liquid water transport through porous transport layers for polymer electrolyte membrane electrolyzers. *Journal of Power Sources*, 437:226910, oct 2019. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2019.226910. URL `https://www.sciencedirect.com/science/article/pii/S0378775319309036`.

[29] M. Sabharwal, L. M. Pant, A. Putz, D. Susac, J. Jankovic, and M. Secanell. Analysis of Catalyst Layer Microstructures: From Imaging to Performance. *Fuel Cells*, 16(6):734–753, 2016. ISSN 16156854.

[30] Mayank Sabharwal, Jeff T. Gostick, and Marc Secanell. Virtual Liquid Water Intrusion in Fuel Cell Gas Diffusion Media. *Journal of The Electrochemical Society*, 165(7):F553–F563, 2018. ISSN 0013-4651.

[31] Pablo A. García-Salaberri, Gisuk Hwang, Marcos Vera, Adam Z. Weber, and Jeff T. Gostick. Effective diffusivity in partially-saturated carbon-fiber gas diffusion layers: Effect of through-plane saturation distribution. *International Journal of Heat and Mass Transfer*, 86:319–333, jul 2015. ISSN 0017-9310.

[32] Iryna V Zenyuk, Adrien Lamibrac, Jens Eller, Dilworth Y Parkinson, Federica Marone, Felix N Bu, and Adam Z Weber. Investigating Evaporation in Gas Diffusion Layers for Fuel Cells with X-ray Computed Tomography. *J. Phys. Chem. C*, 120:8, 2016. doi: 10.1021/acs.jpcc.6b10658. URL `https://pubs.acs.org/sharingguidelines`.

[33] Seyed Mohamad Moosavi, Mathias Niffeler, Jeff Gostick, and Sophia Haussener. Transport characteristics of saturated gas diffusion layers treated with hydrophobic coatings. *Chemical Engineering Science*, 176:503–514, feb 2018. ISSN 0009-2509. doi: 10.1016/J.CES.2017.10.035. URL `https://www.sciencedirect.com/science/article/pii/S0009250917306528`.

[34] A. Arvay, E. Yli-Rantala, C.-H. Liu, X.-H. Peng, P. Koski, L. Cindrella, P. Kauranen, P.M. Wilde, and A.M. Kannan. Characterization techniques for gas diffusion layers for proton exchange membrane fuel cells – A review. *Journal of Power Sources*, 213:317–337, sep 2012. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2012.04.026. URL `https://www.sciencedirect.com/science/article/pii/S0378775312007574`.

[35] Pablo A. García-Salaberri, Iryna V. Zenyuk, Gisuk Hwang, Marcos Vera, Adam Z. Weber, and Jeff T. Gostick. Implications of inherent inhomogeneities in thin carbon fiber-based gas diffusion layers: A comparative modeling study. *Electrochimica Acta*, 295:861–874, feb 2019. ISSN 0013-4686. doi: 10.1016/J.ELECTACTA.2018.09.089. URL `https://www.sciencedirect.com/science/article/pii/S0013468618320759`.

[36] Jeff Gostick, Mahmoudreza Aghighi, James Hinebaugh, Tom Tranter, Michael A Hoeh, — Forschungszentrum, Jülich Harold Day, Brennan Spellacy, Mostafa H Sharqawy, Alan Burns, Werner Lehnert, Forschungszentrum Jülich, Rwth Aachen, and Andreas Putz. OpenPNM: A Pore Network Modeling Package. Technical report.

[37] B Straubhaar, J Pauchet, and M Prat. Pore network modelling of condensation in gas diffusion layers of proton exchange membrane fuel cells. 2016. doi:

10.1016/j.ijheatmasstransfer.2016.06.078. URL `http://dx.doi.org/10.1016/j.ijheatmasstransfer.2016.06.078`.

[38] P.K. Sinha and C.-Y. Wang. Pore-network modeling of liquid water transport in gas diffusion layer of a polymer electrolyte fuel cell. *Electrochimica Acta*, 52 (28):7936–7945, 2007. doi: 10.1016/j.electacta.2007.06.061.

[39] Jin Hyun Nam and Massoud Kaviany. Effective diffusivity and water-saturation distribution in single- and two-layer PEMFC diffusion medium. *International Journal of Heat and Mass Transfer*, 46(24):4595–4611, 2003. ISSN 00179310. doi: 10.1016/S0017-9310(03)00305-3.

[40] Gang Luo, Yan Ji, Chao-Yang Wang, and Puneet K. Sinha. Modeling liquid water transport in gas diffusion layers by topologically equivalent pore network. *Electrochimica Acta*, 55(19):5332–5341, jul 2010. ISSN 0013-4686. doi: 10.1016/J.ELECTACTA.2010.04.078. URL `https://www.sciencedirect.com/science/article/pii/S0013468610006353`.

[41] Hu Dong and Martin J. Blunt. Pore-network extraction from micro-computerized-tomography images. *Phys. Rev. E*, 80:036307, Sep 2009. doi: 10.1103/PhysRevE.80.036307. URL `https://link.aps.org/doi/10.1103/PhysRevE.80.036307`.

[42] Jeff T Gostick. Random Pore Network Modeling of Fibrous PEMFC Gas Diffusion Media Using Voronoi and Delaunay Tessellations. *Journal of The Electrochemical Society*, 160(8):731–743, 2013. doi: 10.1149/2.009308jes. URL `http://jes.ecsdl.org/content/160/8/F731.full.pdf`.

[43] Jeff T. Gostick. Versatile and efficient pore network extraction method using marker-based watershed segmentation. *Physical Review E*, 96(2):1–15, 2017. ISSN 24700053.

[44] Jason Keonhag Lee and Aimy Bazylak. Stochastic Generation of Sintered Titanium Powder-Based Porous Transport Layers in Polymer Electrolyte Membrane Electrolyzers and Investigation of Structural Properties. *ECS Transactions*, 80 (8):1097–1106, aug 2017. ISSN 1938-6737. doi: 10.1149/08008.1097ecst. URL `http://ecst.ecsdl.org/lookup/doi/10.1149/08008.1097ecst`.

[45] Guoqing Wang, Partha P. Mukherjee, and Chao-Yang Wang. Direct numerical simulation (DNS) modeling of PEFC electrodes: Part I. Regular microstructure.

Electrochimica Acta, 51(15):3139–3150, apr 2006. ISSN 0013-4686. doi: 10.1016/J.ELECTACTA.2005.09.002. URL `https://www.sciencedirect.com/science/article/pii/S0013468605010583`.

[46] Partha P Mukherjee and Chao-Yang Wang. Direct Numerical Simulation Modeling of Bilayer Cathode Catalyst Layers in Polymer Electrolyte Fuel Cells. 2007. doi: 10.1149/1.2776221. URL `http://jes.ecsdl.org/content/154/11/B1121.full.pdf`.

[47] Andreas Pfrang, Damien Veyret, Frank Sieker, and Georgios Tsotridis. X-ray computed tomography of gas diffusion layers of PEM fuel cells: Calculation of thermal conductivity. *International Journal of Hydrogen Energy*, 35(8):3751–3757, apr 2010. ISSN 03603199. doi: 10.1016/j.ijhydene.2010.01.085.

[48] Tomas Rosén, Jens Eller, Jinfen Kang, Nikolaos I Prasianakis, John Mantzaras, and Felix N B. Saturation Dependent Effective Transport Properties of PEFC Gas Diffusion Layers. *Journal of The Electrochemical Society*, 159(9):536–544, 2012. doi: 10.1149/2.005209jes.

[49] Pablo A. García-Salaberri, Iryna V. Zenyuk, Andrew D. Shum, Gisuk Hwang, Marcos Vera, Adam Z. Weber, and Jeff T. Gostick. Analysis of representative elementary volume and through-plane regional characteristics of carbon-fiber papers: diffusivity, permeability and electrical/thermal conductivity. *International Journal of Heat and Mass Transfer*, 127:687–703, dec 2018. ISSN 0017-9310.

[50] X. Zhang, Y. Gao, H. Ostadi, K. Jiang, and R. Chen. Modelling water intrusion and oxygen diffusion in a reconstructed microporous layer of pem fuel cells. *International Journal of Hydrogen Energy*, 39(30):17222–17230, 2014. doi: 10.1016/j.ijhydene.2014.08.027. URL `https://www2.scopus.com/inward/record.uri?eid=2-s2.0-84920284788&doi=10.1016%2fj.ijhydene.2014.08.027&partnerID=40&md5=a8ac8f8933b4a99863096f574bf4cf60`. cited By 30.

[51] J. Becker, V. Schulz, and A. Wiegmann. Numerical determination of two-phase material parameters of a gas diffusion layer using tomography images. *Journal of Fuel Cell Science and Technology*, 5(2), 2008. doi: 10.1115/1.2821600. URL `https://www2.scopus.com/inward/record.uri?eid=2-s2.0-44849091925&doi=10.1115%2f1.2821600&partnerID=40&md5=ff4753d8cbae322be338004b036918b1`. cited By 68.

[52] P.K. Sinha, P.P. Mukherjee, and C.-Y. Wang. Impact of gdl structure and wettability on water management in polymer electrolyte fuel cells. *Journal of Materials Chemistry*, 17(30):3089–3103, 2007. doi: 10.1039/b703485g. URL `https://www2.scopus.com/inward/record.uri?eid=2-s2.0-34547424874&doi=10.1039%2fb703485g&partnerID=40&md5=07e57a1282ca6d6427edc7f8872392bc`. cited By 190.

[53] T. Agaesse, A. Lamibrac, F.N. Büchi, J. Pauchet, and M. Prat. Validation of pore network simulations of ex-situ water distributions in a gas diffusion layer of proton exchange membrane fuel cells with x-ray tomographic images. *Journal of Power Sources*, 331:462–474, 2016. doi: 10.1016/j.jpowsour.2016.09.076. URL `https://www2.scopus.com/inward/record.uri?eid=2-s2.0-84991088836&doi=10.1016%2fj.jpowsour.2016.09.076&partnerID=40&md5=fccc7078b51701d5091977c7efc9129d`. cited By 15.

[54] Mayank Sabharwal, Lalit M. Pant, Nilay Patel, and Marc Secanell. Computational Analysis of Gas Transport in Fuel Cell Catalyst Layer under Dry and Partially Saturated Conditions. *Journal of The Electrochemical Society*, 166 (7):F3065–F3080, mar 2019. ISSN 0013-4651. doi: 10.1149/2.0081907jes. URL `http://jes.ecsdl.org/lookup/doi/10.1149/2.0081907jes`.

[55] Volker Paul Schulz, Jürgen Becker, Andreas Wiegmann, Partha P Mukherjee, and Chao-Yang Wang. Modeling of Two-Phase Behavior in the Gas Diffusion Medium of PEFCs via Full Morphology Approach. 2007. doi: 10.1149/1.2472547.

[56] M Tristan Agaesse, M Marc Prat, M Jerome Vicente, Marseille M Yann Bultel, Inp De Grenoble, M Olivier Lottin, Universite De Lorraine, M Joel Pauchet, and Cea Grenoble. *Simulations of one and two-phase flows in porous microstructures, from tomographic images of gas diffusion layers of proton exchange membrane fuel cells.* Phd thesis, Université de Toulouse, 2016. URL `https://oatao.univ-toulouse.fr/17789/`.

[57] J. P. James, H. W. Choi, and J. G. Pharoah. X-ray computed tomography reconstruction and analysis of polymer electrolyte membrane fuel cell porous transport layers. *International Journal of Hydrogen Energy*, 37(23):18216–18230, 2012. ISSN 03603199. doi: 10.1016/j.ijhydene.2012.08.077.

[58] S. Hasanpour, M. Hoorfar, and A.B. Phillion. Different Methods for Determining Porosity of Gas Diffusion Layer using X-ray Microtomography. *Electrochimica Acta*, 185:34–39, dec 2015. ISSN 0013-4686. doi: 10.1016/J.ELECTACTA.2015.10.083. URL `https://www.sciencedirect.com/science/article/pii/S0013468615306630`.

[59] S Odaya, R K Phillips, Y Sharma, J Bellerive, A B Phillion, and & M Hoorfar. X-ray Tomographic Analysis of Porosity Distributions in Gas Diffusion Layers of Proton Exchange Membrane Fuel Cells. doi: 10.1016/j.electacta.2014.11.143.

[60] William K. Epting, Jeff Gelb, and Shawn Litster. Resolving the Three-Dimensional Microstructure of Polymer Electrolyte Fuel Cell Electrodes using Nanometer-Scale X-ray Computed Tomography. *Advanced Functional Materials*, 22(3):555–560, feb 2012. ISSN 1616301X. doi: 10.1002/adfm.201101525. URL `http://doi.wiley.com/10.1002/adfm.201101525`.

[61] S. Litster, W. K. Epting, E. A. Wargo, S. R. Kalidindi, and E. C. Kumbur. Morphological Analyses of Polymer Electrolyte Fuel Cell Electrodes with Nano-Scale Computed Tomography Imaging. *Fuel Cells*, 13(5):n/a–n/a, aug 2013. ISSN 16156846. doi: 10.1002/fuce.201300008. URL `http://doi.wiley.com/10.1002/fuce.201300008`.

[62] Mehdi Andisheh-Tadbir, Francesco P. Orfino, and Erik Kjeang. Three-dimensional phase segregation of micro-porous layers for fuel cells by nano-scale X-ray computed tomography. *Journal of Power Sources*, 310:61–69, apr 2016. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2016.02.001. URL `https://www.sciencedirect.com/science/article/pii/S0378775316301094`.

[63] Pablo A. García-Salaberri, Jeff T. Gostick, Gisuk Hwang, Adam Z. Weber, and Marcos Vera. Effective diffusivity in partially-saturated carbon-fiber gas diffusion layers: Effect of local saturation and application to macroscopic continuum models. *Journal of Power Sources*, 296:440–453, nov 2015. ISSN 0378-7753.

[64] Iryna V Zenyuk, Dilworth Y Parkinson, Gisuk Hwang, and Adam Z Weber. Probing water distribution in compressed fuel-cell gas-diffusion layers using X-ray computed tomography. *Electrochemistry Communications*, 53:24–28, 2015. ISSN 1388-2481. doi: https://doi.org/10.1016/j.elecom.2015.02.005. URL `http://www.sciencedirect.com/science/article/pii/S1388248115000405`.

[65] Simon Thiele, Roland Zengerle, and Christoph Ziegler. Nano-Morphology of a Polymer Electrolyte Fuel Cell Catalyst Layer-Imaging, Reconstruction and Analysis. *Nano Res*, 4(9):849–860, 2011. doi: 10.1007/s12274-011-0141-x. URL `https://www.imtek.de/data/lehrstuehle/app/dokumente/publikationen/publpdf2011/thiele-nano-morphology-of-a-polymer-electrolyte-fuel-cell.pdf`.

[66] Christoph Ziegler, Simon Thiele, and Roland Zengerle. Direct three-dimensional reconstruction of a nanoporous catalyst layer for a polymer electrolyte fuel cell. *Journal of Power Sources*, 196(4):2094–2097, feb 2011. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2010.09.044. URL `https://www.sciencedirect.com/science/article/pii/S0378775310016502`.

[67] Severin Vierrath, Firat Güder, Andreas Menzel, Matthias Hagner, Roland Zengerle, Margit Zacharias, and Simon Thiele. Enhancing the quality of the tomography of nanoporous materials for better understanding of polymer electrolyte fuel cell materials. *Journal of Power Sources*, 285:413–417, jul 2015. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2015.03.110. URL `https://www.sciencedirect.com/science/article/pii/S0378775315005340`.

[68] Xiaoxian Zhang, Hossein Ostadi, Kyle Jiang, and Rui Chen. Reliability of the spherical agglomerate models for catalyst layer in polymer electrolyte membrane fuel cells. *Electrochimica Acta*, 133:475–483, jul 2014. ISSN 0013-4686. doi: 10.1016/J.ELECTACTA.2014.04.060. URL `https://www.sciencedirect.com/science/article/pii/S001346861400807X`.

[69] M. Sabharwal, L. M. Pant, A. Putz, D. Susac, J. Jankovic, and M. Secanell. Analysis of catalyst layer microstructures: From imaging to performance. *Fuel Cells*, 16(6):734–753. doi: 10.1002/fuce.201600008. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/fuce.201600008`.

[70] Mayank Sabharwal, Jeff T. Gostick, and Marc Secanell. Virtual Liquid Water Intrusion in Fuel Cell Gas Diffusion Media. *Journal of The Electrochemical Society*, 165(7):F553–F563, 2018. ISSN 0013-4651. doi: 10.1149/2.0921807jes. URL `http://jes.ecsdl.org/lookup/doi/10.1149/2.0921807jes`.

[71] H. Murakawa, D. Wada, K. Sugimoto, H. Asano, N. Takenaka, and R. Yasuda. Measurements of water distribution in through-plane direction of PEFC by using neutron radiography. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equip-*

ment, 651(1):286–289, sep 2011. ISSN 0168-9002. doi: 10.1016/J.NIMA.2010.12.047. URL `https://www.sciencedirect.com/science/article/pii/S0168900210027889`.

[72] S Chevalier, N Ge, M G George, J Lee, R Banerjee, H Liu, P Shrestha, D Muirhead, J Hinebaugh, Y Tabuchi, T Kotaka, and A Bazylak. Synchrotron X-ray Radiography as a Highly Precise and Accurate Method for Measuring the Spatial Distribution of Liquid Water in Operating Polymer Electrolyte Membrane Fuel Cells. *Journal of The Electrochemical Society*, 164(2):107–114, 2017. doi: 10.1149/2.0041702jes. URL `http://jes.ecsdl.org/content/164/2/F107.full.pdf`.

[73] J. Haußmann, H. Markötter, R. Alink, A. Bauder, K. Dittmann, I. Manke, and J. Scholta. Synchrotron radiography and tomography of water transport in perforated gas diffusion media. *Journal of Power Sources*, 239:611–622, oct 2013. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2013.02.014. URL `https://www.sciencedirect.com/science/article/pii/S0378775313002772`.

[74] Iryna V. Zenyuk, Dilworth Y. Parkinson, Gisuk Hwang, and Adam Z. Weber. Probing water distribution in compressed fuel-cell gas-diffusion layers using X-ray computed tomography. *Electrochemistry Communications*, 53:24–28, apr 2015. ISSN 1388-2481. doi: 10.1016/J.ELECOM.2015.02.005. URL `https://www.sciencedirect.com/science/article/pii/S1388248115000405`.

[75] P Boillat, E H Lehmann, P Trtik, and M Cochet. Neutron imaging of fuel cells – Recent trends and future prospects. *Current Opinion in Electrochemistry*, 5(1):3–10, 2017. ISSN 2451-9103. doi: https://doi.org/10.1016/j.coelec.2017.07.012. URL `http://www.sciencedirect.com/science/article/pii/S2451910317300236`.

[76] I Manke, Ch. Hartnig, M Grünerbel, W Lehnert, N Kardjilov, A Haibel, A Hilger, J Banhart, and H Riesemeier. Investigation of water evolution and transport in fuel cells with high resolution synchrotron x-ray radiography. *Applied Physics Letters*, 90(17):174105, 2007. doi: 10.1063/1.2731440. URL `https://doi.org/10.1063/1.2731440`.

[77] Jacob M. LaManna and Satish G. Kandlikar. Determination of effective water vapor diffusion coefficient in pemfc gas diffusion layers. *International Journal of Hydrogen Energy*, 36(8):5021–5029, apr 2011. ISSN 0360-3199. doi: 10.1016/

J.IJHYDENE.2011.01.036. URL `https://www.sciencedirect.com/science/article/pii/S0360319911000462`.

[78] Grant Unsworth, Lu Dong, and Xianguo Li. Improved Experimental Method for Measuring Gas Diffusivity Through Thin Porous Media. 2012. doi: 10.1002/aic.13911. URL `https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.13911`.

[79] Prafful Mangal, Lalit M. Pant, Nicholas Carrigy, Mark Dumontier, Valentin Zingan, Sushanta Mitra, and Marc Secanell. Experimental study of mass transport in PEMFCs: Through plane permeability and molecular diffusivity in GDLs. *Electrochimica Acta*, 167:160–171, jun 2015. ISSN 0013-4686.

[80] In-Plane Effective Diffusivity in PEMFC Gas Diffusion Layers. *Transport in Porous Media*, 115(3):411–433, dec 2016. ISSN 15731634. doi: 10.1007/s11242-016-0648-4. URL `http://link.springer.com/10.1007/s11242-016-0648-4`.

[81] Hao Xu. Experimental Measurement of Mass Transport Parameters of Gas Diffusion Layer and Catalyst Layer in PEM Fuel Cell. Msc thesis, University of Alberta, 2019.

[82] Jeff T. Gostick, Michael W. Fowler, Mark D. Pritzker, Marios A. Ioannidis, and Leya M. Behra. In-plane and through-plane gas permeability of carbon fiber electrode backing layers. *Journal of Power Sources*, 162(1):228–238, nov 2006. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2006.06.096. URL `https://www.sciencedirect.com/science/article/pii/S0378775306012596`.

[83] Tetsuya Koido, Toru Furusawa, and Koji Moriyama. An approach to modeling two-phase transport in the gas diffusion layer of a proton exchange membrane fuel cell. *Journal of Power Sources*, 175(1):127–136, jan 2008. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2007.09.029. URL `https://www.sciencedirect.com/science/article/pii/S0378775307018915`.

[84] I.S. Hussaini and C.Y. Wang. Measurement of relative permeability of fuel cell diffusion media. *Journal of Power Sources*, 195(12):3830–3840, jun 2010. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2009.12.105. URL `https://www.sciencedirect.com/science/article/pii/S0378775310000339`.

[85] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America A*, 1(6):612, jun 1984. ISSN 1084-7529. doi: 10.1364/josaa.1.000612.

[86] Yi Wu, Saurabh Saxena, Yinjiao Xing, Youren Wang, Chuan Li, Winco Yung, and Michael Pecht. Analysis of Manufacturing-Induced Defects and Structural Deformations in Lithium-Ion Batteries Using Computed Tomography. *Energies*, 11(4):925, apr 2018. ISSN 1996-1073. doi: 10.3390/en11040925. URL `http://www.mdpi.com/1996-1073/11/4/925`.

[87] Anton du Plessis, Chris Broeckhoven, Anina Guelpa, and Stephan Gerhard le Roux. Laboratory x-ray micro-computed tomography: A user guideline for biological samples, jun 2017. ISSN 2047217X.

[88] Stephen Seltzer. Tables of X-Ray Mass Attenuation Coefficients and Mass Energy-Absorption Coefficients, NIST Standard Reference Database 126, 1996. URL `https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients`.

[89] Kleoniki Keklikoglou, Sarah Faulwetter, Eva Chatzinikolaou, Patricia Wils, Jonathan Brecko, Jiří Kvaček, Brian Metscher, and Christos Arvanitidis. Micro-computed tomography for natural history specimens: a handbook of best practice protocols. *European Journal of Taxonomy*, (522):1–55, 2019. ISSN 2118-9773. doi: 10.5852/ejt.2019.522.

[90] F Edward Boas and Dominik Fleischmann. CT artifacts: Causes and reduction techniques. Technical Report 2, 2012. URL `http://www.edboas.com/science/CT/0012.pdf`.

[91] Julia F. Barrett and Nicholas Keat. Artifacts in ct: Recognition and avoidance. *RadioGraphics*, 24(6):1679–1691, 2004. doi: 10.1148/rg.246045065. URL `https://doi.org/10.1148/rg.246045065`. PMID: 15537976.

[92] C Tomasi and R Manduchi. Bilateral Filtering for Gray and Color Images. Technical report.

[93] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji. an open-source platform for biological-image analysis. 9(7):676 – 682, 2012-07. ISSN 1548-7105. Published online 28 June 2012.

[94] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, feb 2000. ISSN 0031-3203.

[95] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979. ISSN 0018-9472.

[96] M Secanell, A Putz, P Wardlaw, V Zingan, M Bhaiya, M Moore, J Zhou, C Balen, and K Domican. OpenFCST: An Open-Source Mathematical Modelling Software for Polymer Electrolyte Fuel Cells.

[97] David A Coker and Salvatore Torquato. Extraction of morphological quantities from a digitized medium . *Journal of Applied Physics*, 77:6087, 1995. doi: 10.1063/1.359134. URL `https://doi.org/10.1063/1.359134`.

[98] C. L. Y. Yeong and S. Torquato. Reconstructing random media. *Phys. Rev. E*, 57:495–506, Jan 1998. doi: 10.1103/PhysRevE.57.495. URL `https://link.aps.org/doi/10.1103/PhysRevE.57.495`.

[99] S Author Torquato and HW Jr Haslach. Random Heterogeneous Materials: Microstructure and Macroscopic Properties. *Applied Mechanics Reviews*, 55(4): B62–B63, 07 2002. ISSN 0003-6900. doi: 10.1115/1.1483342. URL `https://doi.org/10.1115/1.1483342`.

[100] Lalit Mohan Pant. *Stochastic Characterization and Reconstruction of Porous Media*. Ph.d thesis, University of Alberta, 2016.

[101] S Torquato and B Lu. Chord-length distribution function for two-phase random media. Technical Report 4, 1993.

[102] James Hinebaugh, Zachary Fishman, and A Bazylak. Unstructured Pore Network Modeling with Heterogeneous PEMFC GDL Porosity Distributions. *Journal of The Electrochemical Society*, 157:1651–1657, 2010. doi: 10.1149/1.3486095. URL `http://jes.ecsdl.org/content/157/11/B1651.full.pdf`.

[103] Jeff T Gostick, Zohaib A Khan, Thomas G Tranter, Matthew D R Kok, Mehrez Agnaou, Mohammadamin Sadeghi, and Rhodri Jervis. PoreSpy: A Python Toolkit for Quantitative Analysis of Porous Media Images Software • Review • Repository • Archive. 2019.

[104] Andreas Putz, James Hinebaugh, Mahmoudreza Aghighi, Harold Day, Aimy Bazylak, and Jeff Gostick. Introducing OpenPNM: An Open Source Pore Network Modeling Software Package.

[105] D Wilkinson and J F Willemsen. Invasion percolation: a new form of percolation theory. Technical report, 1983. URL `https://iopscience.iop.org/article/10.1088/0305-4470/16/14/028/pdf`.

[106] J. Hoshen and R. Kopelman. Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm. *Phys. Rev. B*, 14:3438–3445, Oct 1976.

[107] P A García-Salaberri, G Hwang, M Vera, A Z Weber, J T Gostick, and Jeff T Gostick. Supplementary material to Effective diffusivity in partially-saturated carbon-fiber gas diffusion layers: Effect of through-plane saturation distribution. URL `https://ars.els-cdn.com/content/image/1-s2.0-S001793101500246X-mmc1.pdf`.

[108] Rinat R Rashapov, Jonathan Unno, and Jeff T Gostick. Characterization of PEMFC Gas Diffusion Layer Porosity. *Journal of The Electrochemical Society*, 162(6):603–612, 2015. doi: 10.1149/2.0921506jes. URL `http://jes.ecsdl.org/content/162/6/F603.full.pdf`.

[109] S Shukla, F Wei, M Mandal, J Zhou, M S Saha, J Stumper, and M Secanell. Determination of PEFC Gas Diffusion Layer and Catalyst Layer Porosity Utilizing Archimedes Principle. *Journal of The Electrochemical Society*, 166(15): 1142–1147, 2019. doi: 10.1149/2.0251915jes.

[110] Hechao Li, Somya Singh, Nikhilesh Chawla, and Yang Jiao. Direct extraction of spatial correlation functions from limited x-ray tomography data for microstructural quantification. *Materials Characterization*, 140:265–274, jun 2018. ISSN 10445803. doi: 10.1016/j.matchar.2018.04.020.

[111] Sean P. Rigby and Lynn F. Gladden. Deconvolving pore shielding effects in mercury porosimetry data using NMR techniques. *Chemical Engineering Science*, 55(23):5599–5612, dec 2000. ISSN 00092509. doi: 10.1016/S0009-2509(00)00192-5.

[112] L. Cindrella, A.M. Kannan, J.F. Lin, K. Saminathan, Y. Ho, C.W. Lin, and J. Wertz. Gas diffusion layer for proton exchange membrane fuel cells—A review. *Journal of Power Sources*, 194(1):146–160, oct 2009. ISSN 0378-7753.

[113] A review of gas diffusion layer in PEM fuel cells: Materials and designs. *International Journal of Hydrogen Energy*, 37(7):5850–5865, apr 2012. ISSN 0360-3199.

[114] Timothy Sauer. *Numerical analysis*, volume 9. 2006. ISBN 9780321783677. doi: 10.1145/365719.366369.

[115] Jeff T. Gostick, Marios A. Ioannidis, Michael W. Fowler, and Mark D. Pritzker. Wettability and capillary behavior of fibrous gas diffusion media for polymer electrolyte membrane fuel cells. *Journal of Power Sources*, 194(1):433–444, oct 2009. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2009.04.052. URL `https://www.sciencedirect.com/science/article/pii/S0378775309007381`.

[116] F. Arbabi, H. Montazeri, R. Abouatallah, R. Wang, and A. Bazylak. Three-Dimensional Computational Fluid Dynamics Modelling of Oxygen Bubble Transport in Polymer Electrolyte Membrane Electrolyzer Porous Transport Layers. *Journal of The Electrochemical Society*, 163(11):F3062–F3069, 2016. ISSN 0013-4651. doi: 10.1149/2.0091611jes. URL `http://jes.ecsdl.org/lookup/doi/10.1149/2.0091611jes`.

[117] Mayank Sabharwal, Andreas Michael Vincent Putz, Darija Susac, Jasna Jankovic, and Marc Secanell. Improving fib-sem reconstructions by using epoxy resin embedding. 77(11):1337–1349, 2017. doi: 10.1149/07711.1337ecst.

[118] J Lobato, Ae P Cañizares, Ae M A Rodrigo, Ae C Ruiz-López, and Ae J J Linares. Influence of the Teflon loading in the gas diffusion layer of PBI-based PEM fuel cells. *Journal of Applied Electrochemistry*, 38(6):793–802, 2008. doi: 10.1007/s10800-008-9512-8. URL `https://link.springer.com/content/pdf/10.1007{%}2Fs10800-008-9512-8.pdf`.

[119] Nada Zamel, Nelson G.C. Astrath, Xianguo Li, Jun Shen, Jianqin Zhou, Francine B.G. Astrath, Haijiang Wang, and Zhong-Sheng Liu. Experimental measurements of effective diffusion coefficient of oxygen–nitrogen mixture in PEM fuel cell diffusion media. *Chemical Engineering Science*, 65(2):931–937, jan 2010. ISSN 0009-2509. doi: 10.1016/J.CES.2009.09.044. URL `https://www.sciencedirect.com/science/article/pii/S0009250909006472`.

[120] Reto Flückiger, Stefan A. Freunberger, Denis Kramer, Alexander Wokaun, Günther G. Scherer, and Felix N. Büchi. Anisotropic, effective diffusivity of porous gas diffusion layer materials for PEFC. *Electrochimica Acta*, 54(2):551–559, dec 2008. ISSN 0013-4686. doi: 10.1016/J.ELECTACTA.

2008.07.034. URL `https://www.sciencedirect.com/science/article/pii/S0013468608008864`.

[121] Nicholas B. Carrigy, Lalit M. Pant, Sushanta Mitra, and Marc Secanell. Knudsen diffusivity and permeability of pemfc microporous coated gas diffusion layers for different polytetrafluoroethylene loadings. *Journal of The Electrochemical Society*, 160(2):F81–F89, 2013. doi: 10.1149/2.036302jes. URL `http://jes.ecsdl.org/content/160/2/F81.abstract`.

[122] A. Tamayol, F. McGregor, and M. Bahrami. Single phase through-plane permeability of carbon paper gas diffusion layers. *Journal of Power Sources*, 204:94 – 99, 2012. ISSN 0378-7753. doi: https://doi.org/10.1016/j.jpowsour.2011.11.084. URL `http://www.sciencedirect.com/science/article/pii/S037877531102372X`.

[123] Hiroshi Ito, Katsuya Abe, Masayoshi Ishida, Akihiro Nakano, Tetsuhiko Maeda, Tetsuo Munakata, Hironori Nakajima, and Tatsumi Kitahara. Effect of through-plane distribution of polytetrafluoroethylene in carbon paper on in-plane gas permeability. *Journal of Power Sources*, 248:822–830, feb 2014. ISSN 0378-7753. doi: 10.1016/J.JPOWSOUR.2013.10.009. URL `https://www.sciencedirect.com/science/article/pii/S0378775313016601`.

[124] Westgrid. `https://www.westgrid.ca/`.

[125] Computecanada. `https://www.computecanada.ca/`.

Appendices

Appendix A

PNM code for MIP and dry transport simulations

```
# −∗− coding: utf−8 −∗−
”””
Created on Wed Jul 25 12:48:35 2018

@author: Yeop
”””
from porespy.networks import regions_to_network, add_boundary_regions
from porespy.networks import label_boundary_cells
from porespy.tools import pad_faces
from porespy.tools import make_contiguous
import porespy as ps
import openpnm as op
import matplotlib.pyplot as plt
import os
import imageio
import scipy as sp
import time
import numpy as np
from openpnm.models import physics as pm
from tqdm import tqdm
import skimage.external.tifffile as tiff

start = time.time()
########################### Import image
    ↪ ################################

ws = op.Workspace()
ws.clear()
prj = op.Project(name = ’reconstruction’)

path = ’/home/yeop/Documents/Article/Ti/1/’# + str(i+1)+’/’
# path = os.getcwd()
file = ’stack1.tif’
fetch_file = os.path.join(path,file)
im = imageio.mimread(fetch_file, memtest=False)
im = sp.array(im, dtype = bool)
im = sp.invert(im)
voxel = 1.79e−6

########################### SNOW algorithm
    ↪ ################################


regions = ps.filters.snow_partitioning(im, dt = None, r_max=5, sigma=0.4, return_all=True,
                         mask=True, randomize=True)
```

```
im = regions.im
dt = regions.dt
regions = regions.regions
b_num = sp.amax(regions)

plt.imshow((regions*im)[10, :, :], cmap=plt.cm.nipy_spectral)
plt.imshow((im)[10, :, :], cmap=plt.cm.Greys)
plt.axis(’off’)
plt.show()

structure_size = im.shape
porosity = ps.metrics.porosity(im)

# Boundary pores
regions = add_boundary_regions(regions=regions, faces=[’front’, ’back’, ’left’,
                                            ’right’, ’top’, ’bottom’])


# Padding distance transform and image to extract geometrical properties
dt = pad_faces(im=dt, faces=[’front’, ’back’, ’left’,
                                            ’right’, ’top’, ’bottom’])
im = pad_faces(im=im, faces=[’front’, ’back’, ’left’,
                                            ’right’, ’top’, ’bottom’])
regions = regions*im
regions = make_contiguous(regions)

plt.imshow((regions)[:, 13, :], cmap=plt.cm.nipy_spectral)
plt.axis(’off’)
plt.show()


# Extract void and throat information from image
net = regions_to_network(im=regions, dt=dt, voxel_size=voxel)
# Find void to void connections of boundary and internal voids
boundary_labels = net[’pore.label’] > b_num
loc1 = net[’throat.conns’][:, 0] < b_num
loc2 = net[’throat.conns’][:, 1] >= b_num
pore_labels = net[’pore.label’] <= b_num
loc3 = net[’throat.conns’][:, 0] < b_num
loc4 = net[’throat.conns’][:, 1] < b_num
net[’pore.boundary’] = boundary_labels
net[’throat.boundary’] = loc1 * loc2
net[’pore.internal’] = pore_labels
net[’throat.internal’] = loc3 * loc4

# label boundary cells
net = label_boundary_cells(network=net, boundary_faces=[’front’, ’back’, ’left’,
                                            ’right’, ’top’, ’bottom’])

############################## Calculate porosity
    ↪ ##############################

porosity_network = sum(net[’pore.volume’][net[’pore.internal’]])/(structure_size[0]*structure_size[1]*structure_size
    ↪ [2]*voxel**3)
print(”porosity_image_:”, porosity, ”porosity_network_:”, porosity_network)
############################## Define Network
    ↪ ##################################
# print(net.keys())
pn = op.network.GenericNetwork(project = prj, name = ’network’)
pn.update(net)
#print(pn)
pore_d = pn[’pore.diameter’][pn.pores(’internal’)]
pore_r = (pn[’pore.diameter’][pn.pores(’internal’)])/2
pore_v = pn[’pore.volume’][pn.pores(’internal’)]
pore_r_eq = (pn[’pore.equivalent_diameter’][pn.pores(’internal’)])/2

############################## Add geometrical model
```

```
    ↪ ##########################

net_health = pn.check_network_health()
if len(net_health['trim_pores']) > 0:
    op.topotools.trim(network=pn, pores=net_health['trim_pores'])


############################### Define phases
    ↪ ###############################
hg = op.phases.Mercury(network=pn, name = 'hg')
air = op.phases.Air(network=pn, name = 'air')
water = op.phases.Water(network = pn, name = 'water')
air['pore.temperature'] = 353
water['pore.temperature'] = 353
air['pore.diffusivity']= 2.2e-5
air['throat.diffusivity'] = 2.2e-5
air['pore.molar_density'] = 34.52
water['pore.contact_angle'] = 110
water['pore.surface_tension'] = 0.0725

############################### Define physics
    ↪ ##############################
phys_hg = op.physics.Standard(network=pn, phase=hg, geometry = pn, name = 'phys_hg')
phys_air = op.physics.Standard(network = pn, phase = air, geometry = pn, name = 'phys_air')
phys_water = op.physics.Standard(network = pn, phase = water, geometry = pn, name = 'phys_water')

throat_diam = 'throat.diameter'
pore_diam = 'pore.indiameter'

pmod = pm.capillary_pressure.washburn
phys_hg.add_model(propname='throat.entry_pressure',
                     model=pmod,
                     surface_tension='pore.surface_tension',
                     contact_angle='pore.contact_angle',
                     diameter='throat.diameter')

phys_air.add_model(propname='throat.entry_pressure',
                   model=pmod,
                   surface_tension='pore.surface_tension',
                     contact_angle='pore.contact_angle',
                     diameter='throat.diameter')
phys_water.add_model(propname='throat.entry_pressure',
                     model=pmod,
                     surface_tension='pore.surface_tension',
                     contact_angle='pore.contact_angle',
                     diameter='throat.diameter')



####### ADD LATE PORE FILLING
phys_hg.add_model(propname='pore.pc_star',
                  model=op.models.misc.from_neighbor_throats,
                  throat_prop='throat.entry_pressure',
                  mode='min')
phys_hg.add_model(propname = 'pore.late_filling',
                  model = op.models.physics.multiphase.late_filling,
                  pressure = 'pore.pressure',
                  Pc_star = 'pore.pc_star',
                  eta = 2.5, Swp_star = 0.25,
                  regen_mode = 'deferred')
phys_hg['throat.pc_star'] = phys_hg['throat.entry_pressure']
phys_hg.add_model(propname = 'throat.late_filling',
                  model = op.models.physics.multiphase.late_filling,
                  Pc_star = 'throat.pc_star',
                  pressure = 'throat.pressure',
                  eta = 2.5, Swp_star = 0.25,
                  regen_mode = 'deferred')
```

```
lpf = ’pore.late_filling’
phys_water.add_model(propname=’pore.pc_star’,
                     model=op.models.misc.from_neighbor_throats,
                     throat_prop=’throat.entry_pressure’,
                     mode=’min’)
phys_water.add_model(propname=lpf,
                     model=pm.multiphase.late_filling,
                     pressure=’pore.pressure’,
                     Pc_star=’pore.pc_star’,
                     Swp_star=0.25,
                     eta=2.5)

Sat=[]
################################ run MIP simulation
mip = op.algorithms.Porosimetry(network=pn, name = ’MIP’)
mip.setup(phase=hg)
mip.set_inlets(pores=pn.pores(’boundary’))
mip.settings[’late_pore_filling’] = ’pore.late_filling’
mip.run(points = 100)
for pc in tqdm(np.unique(mip[’pore.invasion_pressure’])):
    # Update the occupancy of each phase at the current value of s
    hg.update(mip.results(Pc=pc))
    # The following is kludgey. We have added a Multiphase class that does
    # this automatically but I won’t use it here for simplicity
    air[’pore.occupancy’] = 1−hg[’pore.occupancy’]
    air[’throat.occupancy’] = 1−hg[’throat.occupancy’]
    # Regenerate the physics’ to remove saturation effects from previous loops
    phys_air.regenerate_models()
    phys_water.regenerate_models()
    this_sat = 0
    this_sat += np.sum(pn[”pore.volume”][hg[”pore.occupancy”] == 1])
    this_sat += np.sum(pn[”throat.volume”][hg[”throat.occupancy”] == 1])
    Sat.append(this_sat)
    op.io.VTK.save(network = pn, phases=[hg], filename=path+’MIP/hg_mip_’+str(len(Sat)))


x,y = mip.get_intrusion_data()
mip.plot_intrusion_curve()

###### Output MIP data
np.savetxt(path+’MIP/sat.csv’, Sat/Sat[−1], delimiter=’,’)
np.savetxt(path+”pnm_mip.csv”, (x,y), delimiter = ’,’)

############################# Add geometry models
pn.add_model(propname=’throat.endpoints’,
             model=op.models.geometry.throat_endpoints.spherical_pores,
             pore_diameter=’pore.inscribed_diameter’,
             throat_diameter=’throat.inscribed_diameter’,
             throat_centroid=’throat.centroid’)
pn.add_model(propname=’throat.conduit_lengths’,
             model=op.models.geometry.throat_length.conduit_lengths)
pn.add_model(propname=’pore.area’,
             model=op.models.geometry.pore_area.sphere,
             pore_diameter=’pore.equivalent_diameter’)
pn.add_model(propname = ’throat.area’,
             model = op.models.geometry.throat_area.cylinder,
             throat_diameter=’throat.equivalent_diameter’
             )
pn.add_model(propname = ’pore.diameter’,
             model = op.models.geometry.pore_size.equivalent_diameter)
pn.add_model(propname = ’throat.diameter’,
             model = op.models.geometry.throat_size.equivalent_diameter)



phys_air.add_model(propname = ’throat.diffusive_conductance’,
                   model = op.models.physics.diffusive_conductance.ordinary_diffusion,
                   pore_area=’pore.area’,
                   throat_area=’throat.area’,
```

```
                    pore_diffusivity='pore.diffusivity',
                    throat_diffusivity='throat.diffusivity',
                    conduit_lengths='throat.conduit_lengths',
                    conduit_shape_factors='throat.poisson_shape_factors')
phys_air.add_model(propname = 'throat.hydraulic_conductance',
                    model = op.models.physics.hydraulic_conductance.hagen_poiseuille,
                    pore_area='pore.area',
                    throat_area='throat.area',
                    pore_viscosity='pore.viscosity',
                    throat_viscosity='throat.viscosity',
                    conduit_lengths='throat.conduit_lengths',
                    conduit_shape_factors='throat.flow_shape_factors')

######################### run diffusivity simulation
    ↪ #########################
def diffusivity(phase, conductance_model, axis=0):
    fd = op.algorithms.FickianDiffusion(network=pn)
    fd.setup(phase=phase, conductance=conductance_model)
    if axis == 0:
        fd.set_value_BC(pores=pn.pores('front'), values=0.4)
        fd.set_value_BC(pores=pn.pores('back'), values=0.01)
        L = im.shape[1]*voxel
        A = im.shape[0]*im.shape[2]*(voxel**2)
    elif axis == 1:# Through plane
        fd.set_value_BC(pores=pn.pores('left'), values=0.4)
        fd.set_value_BC(pores=pn.pores('right'), values=0.01)
        L = im.shape[0]*voxel
        A = im.shape[1]*im.shape[2]*(voxel**2)
    else:
        fd.set_value_BC(pores=pn.pores('top'), values=0.4)
        fd.set_value_BC(pores=pn.pores('bottom'), values=0.01)
        L = im.shape[2]*voxel
        A = im.shape[0]*im.shape[1]*(voxel**2)
    fd.run()
    return fd.calc_effective_diffusivity(domain_area=A, domain_length=L)
prop = 'throat.diffusive_conductance'

Deff_0 = diffusivity(air, prop, axis = 0)/air['pore.diffusivity'][0]
Deff_1 = diffusivity(air, prop, axis = 1)/air['pore.diffusivity'][0]
Deff_2 = diffusivity(air, prop, axis = 2)/air['pore.diffusivity'][0]

print('Diffusibility')
print('X␣:␣', Deff_0)
print('Y␣:␣', Deff_1)
print('Z␣:␣', Deff_2)


######################## run permeability simulation
    ↪ #########################
def permeability(phase, conductance_model, axis=0):
    perm = op.algorithms.StokesFlow(network=pn)
    perm.setup(phase=phase, conductance=conductance_model)
    if axis == 0:
        perm.set_value_BC(pores=pn.pores('front'), values=1)
        perm.set_value_BC(pores=pn.pores('back'), values=0.01)
        L = im.shape[1]*voxel
        A = im.shape[0]*im.shape[2]*(voxel**2)
    elif axis == 1: # Through plane
        perm.set_value_BC(pores=pn.pores('left'), values=1)
        perm.set_value_BC(pores=pn.pores('right'), values=0.01)
        L = im.shape[0]*voxel
        A = im.shape[1]*im.shape[2]*(voxel**2)
    else:
        perm.set_value_BC(pores=pn.pores('top'), values=1)
        perm.set_value_BC(pores=pn.pores('bottom'), values=0.01)
        L = im.shape[2]*voxel
        A = im.shape[0]*im.shape[1]*(voxel**2)
    perm.run()
```

```
    return perm.calc_effective_permeability(domain_area=A, domain_length=L)
prop = ’throat.hydraulic_conductance’

perm_0 = permeability(air, prop, axis = 0)
perm_1 = permeability(air, prop, axis = 1)
perm_2 = permeability(air, prop, axis = 2)

print(’Permeability’)
print(’X␣:␣’, perm_0)
print(’Y␣:␣’, perm_1)
print(’Z␣:␣’, perm_2)


############################### Rel diff
    ↪ ###############################
# Ignore the pore entry pressures
phys_air[’pore.entry_pressure’] = −999999
phys_water[’pore.entry_pressure’] = −999999



############################### Plot PSD (dv/v)
    ↪ ###############################
v_tot = pore_v.sum()
no_bins = 100
n, bins, patches = plt.hist(pore_r, bins = no_bins)
x_new = []
for x in range(len(bins)):
    if x == len(bins)−1:
        break
    else:
        x_new.append((bins[x]+bins[x+1])/2)


idx = []
dV = []
for x in range(len(bins)−1):
    if x == 99:
        idx.append(np.where((pore_r >= bins[x]) & (pore_r <= bins[x+1])))
    else:
        idx.append(np.where((pore_r >= bins[x]) & (pore_r < bins[x+1])))
    dV.append(pore_v[idx[x]].sum())

dv_v = dV/v_tot
plt.figure()
plt.plot(x_new, dv_v)
plt.show()
np.savetxt(path +”pnm_psd_pore_inscribed.csv”, (x_new, dv_v,), delimiter = ’,’)
n, bins, patches = plt.hist(pore_r_eq, bins = no_bins)


timestamp = time.strftime(”%Y−%m−%d_%H−%M−%S”)
# take the current date and time
############################### Document gen
    ↪ ##################################
def DocumentationCreator(timestamp):
    ”””
    Creates a text file within the same folder as the image stack that is used to
    document properties of the generated strucutre

    Args:
        − folder_name (str, class variable): folder path for where images will be saved
        − timestamp (str): string containing information of the date and time the
            structure was generated
        − other class variables are included to be added to the documentation text

    Returns:
        − text document containing parameters describing the generated structure
```

```
    """


    por = porosity

    folder_name = path
    # imports folder name
    document_name = folder_name + '␣documentation'
    document_file = open(folder_name + '/Void_test_eq''.txt','w')
    # creates text file in same folder as image stack

    document_file.write('%s\n\n' %(document_name))
    document_file.write('Date␣generated:␣%s\n' %(timestamp))
    # saves date and time structure was generated
    document_file.write('Structure␣size:␣%s\n' %(str(structure_size)))
    # saves structure size
    document_file.write('Image␣porosity:␣%.10f\n' %(por))
    # saves actual porosity of generated structure
    document_file.write('Network␣porosity:␣%.10f\n' %(porosity_network))
    # saves extracted porosity of generated structure
    document_file.write('Diffusibility␣x␣:␣%.10f\n' %(Deff_1))
    document_file.write('Diffusibility␣y␣:␣%.10f\n' %(Deff_0))
    document_file.write('Diffusibility␣z␣:␣%.10f\n' %(Deff_2))
    document_file.write('Permeability␣x␣:␣%.20f\n' %(perm_1))
    document_file.write('Permeability␣y␣:␣%.20f\n' %(perm_0))
    document_file.write('Permeability␣z␣:␣%.20f\n' %(perm_2))

    document_file.close()

    return
################################ Export data
    ↪ ################################
DocumentationCreator(timestamp)

por_x = ps.metrics.porosity_profile(im, axis = 0)
por_y = ps.metrics.porosity_profile(im, axis = 1)
por_z = ps.metrics.porosity_profile(im, axis = 2)

ws.save_project(prj, path+'recon_project_ins')
prj.export_data(phases = [air, hg, water, mip], filename=path + 'PNM_dry', filetype='vtp')
ps.io.to_vtk(im, path=path + '/extracted_network', voxel_size=voxel)

############################ porosity profile
    ↪ ################################
plt.figure(0, figsize = (13,3))
for i in range (1, 4):
    plt.subplot(1, 3, i)
plt.subplot(1,3,1)
plt.plot(por_x)

plt.title(r'X-direction')
plt.ylabel(r'Porosity(%)')
plt.subplot(1,3,2)
plt.plot(por_y)

plt.title(r'Y-direction')
plt.ylabel(r'Porosity(%)')
plt.subplot(1,3,3)
plt.plot(por_z)

plt.title('Z-direction')
plt.ylabel('Porosity(%)')

plt.tight_layout()
plt.savefig(path + "prosity_profile_TIfoam")

plt.show()
```

```
end = time.time()
print(”Time␣elapsed␣wall−clock␣time␣:␣”, end−start)
```

Appendix B

PNM code for relative transport simulations

```
import openpnm as op
import numpy as np
import matplotlib.pyplot as plt
import os
import scipy as sp
import imageio
import csv
from tqdm import tqdm


path = ’/home/yeop/Documents/Article/Ti/1/’
wrk = op.Workspace()
wrk.settings[’loglevel’] = 50
print(wrk)

file = ’stack1.tif’
fetch_file = os.path.join(path,file)
im = imageio.mimread(fetch_file)
im = sp.array(im, dtype = bool)
im = sp.invert(im)
voxel = 1.79e−6

# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−− Invasion percolation
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−

# f = [0.1, 0.3, 0.5, 0.7, 1] # When testing inlet conditions,
# for i in f:
f = 1
wrk.clear()
wrk.load_project(path + ’recon_project_eq’)
```

```
prj = wrk[’reconstruction’]
pn = prj.network
air = prj.phases(’air’)
water = prj.phases(’water’)
phys_air = prj.physics(’phys_air’)
phys_water = prj.physics(’phys_water’)

water[”pore.occupancy”] = False
water[”throat.occupancy”] = False
air[”pore.occupancy”] = False
air[”throat.occupancy”] = False

phys_air.add_model(propname=’throat.multiphase_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.diffusive_conductance’,
              mode=’medium’,
              factor=1e−8)
phys_water.add_model(propname=’throat.multiphase_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.diffusive_conductance’,
              mode=’medium’,
              factor=1e−8)
phys_air.add_model(propname=’throat.conduit_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.hydraulic_conductance’,
              mode=’medium’,
              factor=1e−8)
phys_water.add_model(propname=’throat.conduit_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.hydraulic_conductance’,
              mode=’medium’,
              factor=1e−8)

#for CL Inlet$
# inlet = np.where(pn[’pore.diameter’]<=4.5e−9)
#for PTL Inlet$
inlet = np.random.permutation(pn.pores(’left’))[:int(pn.num_pores(’left’)∗f)]

# Instantiate an invasion percolation algorithm and inject water from the top
ip = op.algorithms.InvasionPercolation(network=pn)
ip.setup(phase=water)
ip.set_inlets(pores=inlet)
ip.run()

# Instantiate a diffusion algorithm for the air phase
fda_tp = op.algorithms.FickianDiffusion(network=pn, phase=air)
# Apply a concentration gradient from top to bottom, same direction as invasion
fda_tp.set_value_BC(pores=pn.pores(’left’), values=1)
fda_tp.set_value_BC(pores=pn.pores(’right’), values=0)
# Run it to obtain the effective diffusivity of the dry network
fda_tp.run()
Deff_a_tp = fda_tp.rate(pores=pn.pores(’left’))
# Instantiate a diffusion algorithm for the water phase
fda_ip = op.algorithms.FickianDiffusion(network=pn, phase=air)
fda_ip.set_value_BC(pores=pn.pores(’top’), values=1)
fda_ip.set_value_BC(pores=pn.pores(’bottom’), values=0)

fda_ip.run()
Deff_a_ip = fda_ip.rate(pores=pn.pores(’top’))

sta_ip = op.algorithms.StokesFlow(network = pn, phase = air)
sta_ip.set_value_BC(pores = pn.pores(’top’), values = 1)
sta_ip.set_value_BC(pores = pn.pores(’bottom’), values = 0)
sta_ip.run()
K_a_ip = sta_ip.rate(pn.pores(’top’))
phys_air.regenerate_models()

sta_tp = op.algorithms.StokesFlow(network = pn, phase = air)
```

```
sta_tp.set_value_BC(pores = pn.pores(’left’), values = 1)
sta_tp.set_value_BC(pores = pn.pores(’right’), values = 0)
sta_tp.run()
K_a_tp = sta_tp.rate(pn.pores(’left’))

stw_tp = op.algorithms.StokesFlow(network = pn, phase = water)
stw_tp.set_value_BC(pores = pn.pores(’right’), values = 1)
stw_tp.set_value_BC(pores = pn.pores(’left’), values = 0)
stw_tp.run()
K_w_tp = stw_tp.rate(pn.pores(’right’))

swat = []
deffa_tp = []
deffa_ip = []
ka_tp = []
kw_tp = []
pc = []

fda_tp.settings[’conductance’] = ’throat.multiphase_conductance’
fda_ip.settings[’conductance’] = ’throat.multiphase_conductance’
sta_tp.settings[’conductance’] = ’throat.conduit_conductance’
stw_tp.settings[’conductance’] = ’throat.conduit_conductance’

# Loop over 40 invasion configurations
for seq in tqdm(sp.linspace(0, pn.Nt+pn.Np,40)):
    # Update the occupancy of each phase at the current value of s
    water.update(ip.results(Snwp=seq/(pn.Np+pn.Nt)))
    # The following is kludgey. We have added a Multiphase class that does
    # this automatically but I won’t use it here for simplicity
    air[’pore.occupancy’] = ~water[’pore.occupancy’]
    air[’throat.occupancy’] = ~water[’throat.occupancy’]
    # Regenerate the physics’ to remove saturation effects from previous loops
    phys_air.regenerate_models()
    phys_water.regenerate_models()
    # Run both algorithms
    fda_tp.run()
    fda_ip.run()
    sta_tp.run()
    stw_tp.run()
    # Determine the flux into the top of the domain, in each phase
    deffa_tp.append(fda_tp.calc_effective_diffusivity(domain_length = im.shape[0]*voxel, domain_area = im.shape[1]*
        ↪ im.shape[2]*voxel**2 ))
    deffa_ip.append(fda_ip.calc_effective_diffusivity(domain_length = im.shape[1]*voxel, domain_area = im.shape[0]*
        ↪ im.shape[2]*voxel**2))
    ka_tp.append(sta_tp.calc_effective_permeability(domain_length = im.shape[0]*voxel, domain_area = im.shape[1]*
        ↪ im.shape[2]*voxel**2))
    kw_tp.append(stw_tp.calc_effective_permeability(domain_length = im.shape[0]*voxel, domain_area = im.shape
        ↪ [1]*im.shape[2]*voxel**2))

    swat.append(seq/(pn.Np+pn.Nt))
    op.io.VTK.save(network = pn, phases=[fda_tp,fda_ip,sta_tp,sta_ip,stw_tp,air, water, phys_air, phys_water],
        ↪ filename=path +’any/ip/water_ip_’+str(int(i*10))+ ’_’ +str(len(swat)))
    if water[’throat.occupancy’].any():
        pi = np.max(ip[’throat.entry_pressure’][water[’throat.occupancy’]])
    else:
        pi = 0
    pc.append(pi)

# Plot the results for each phase as function of water saturation
deffa_tp_IP = deffa_tp/deffa_tp[0]
deffa_ip_IP = deffa_ip/deffa_ip[0]
ka_tp_IP = ka_tp/ka_tp[0]
kw_tp_IP = kw_tp/kw_tp[-1]

figure_0 = plt.plot(pc, swat)
plt.show()
np.savetxt(path+”pnm_water_ip.csv”, (pc,swat), delimiter = ’,’)
with open(path+’pnm_rel_IP_’+str(int(i*100))+’.csv’, ’w’) as csv_file:
```

```
    csv_writer = csv.writer(csv_file, delimiter = ’,’)
    csv_writer.writerow(swat)
    csv_writer.writerow(np.asarray(deffa_tp_IP).flatten())
    csv_writer.writerow(np.asarray(deffa_ip_IP).flatten())
    csv_writer.writerow(np.asarray(ka_tp_IP).flatten())
    csv_writer.writerow(np.asarray(kw_tp_IP).flatten())
fig = plt.plot(swat, deffa_tp_IP, ’b−o’)
fig = plt.plot(swat, deffa_ip_IP, ’r−o’)
fig = plt.show()
fig1 = plt.plot(swat,ka_tp_IP, ’b−o’)
fig1 = plt.plot(swat,kw_tp_IP, ’r−o’)
fig1 = plt.show()

# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−− Ordinary percolation
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

wrk.clear()
wrk.load_project(path + ’recon_project_eq’)
prj = wrk[’reconstruction’]
pn = prj.network
air = prj.phases(’air’)
water = prj.phases(’water’)
phys_air = prj.physics(’phys_air’)
phys_water = prj.physics(’phys_water’)

water[”pore.occupancy”] = False
water[”throat.occupancy”] = False
air[”pore.occupancy”] = False
air[”throat.occupancy”] = False

phys_air.add_model(propname=’throat.multiphase_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.diffusive_conductance’,
              mode=’medium’,
              factor=1e−8)
phys_water.add_model(propname=’throat.multiphase_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.diffusive_conductance’,
              mode=’medium’,
              factor=1e−8)
phys_air.add_model(propname=’throat.conduit_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.hydraulic_conductance’,
              mode=’medium’,
              factor=1e−8)
phys_water.add_model(propname=’throat.conduit_conductance’,
              model=op.models.physics.multiphase.conduit_conductance,
              throat_conductance=’throat.hydraulic_conductance’,
              mode=’medium’,
              factor=1e−8)

# Instantiate an ordinary percolation algorithm and inject water in through−plane
op1 = op.algorithms.OrdinaryPercolation(network=pn)
```

```
op1.setup(phase=water)
op1.set_inlets(pores=inlet)
op1.run(points = 70)

# Instantiate a diffusion algorithm for the air phase
fda_tp = op.algorithms.FickianDiffusion(network=pn, phase=air)
# Apply a concentration gradient from top to bottom, same direction as invasion
fda_tp.set_value_BC(pores=pn.pores(’left’), values=1)
fda_tp.set_value_BC(pores=pn.pores(’right’), values=0)
# Run it to obtain the effective diffusivity of the dry network
fda_tp.run()
Deff_a_tp = fda_tp.rate(pores=pn.pores(’left’))
# Instantiate a diffusion algorithm for the water phase
fda_ip = op.algorithms.FickianDiffusion(network=pn, phase=air)
# Apply a concentration gradient from top to bottom, same direction as invasion
fda_ip.set_value_BC(pores=pn.pores(’top’), values=1)
fda_ip.set_value_BC(pores=pn.pores(’bottom’), values=0)

fda_ip.run()
Deff_a_ip = fda_ip.rate(pores=pn.pores(’top’))

sta_tp = op.algorithms.StokesFlow(network = pn, phase = air)
sta_tp.set_value_BC(pores = pn.pores(’left’), values = 1)
sta_tp.set_value_BC(pores = pn.pores(’right’), values = 0)
sta_tp.run()
K_a_tp = sta_tp.rate(pn.pores(’left’))

stw_tp = op.algorithms.StokesFlow(network = pn, phase = water)
stw_tp.set_value_BC(pores = pn.pores(’right’), values = 1)
stw_tp.set_value_BC(pores = pn.pores(’left’), values = 0)
stw_tp.run()
K_w_tp = stw_tp.rate(pn.pores(’right’))

deffa_tp = []
deffa_ip = []
ka_tp = []
kw_tp = []

# Here we must tell the transport alg to use the multiphase conductance, rather
# than the default which is the single phase value
fda_tp.settings[’conductance’] = ’throat.multiphase_conductance’
fda_ip.settings[’conductance’] = ’throat.multiphase_conductance’
sta_tp.settings[’conductance’] = ’throat.conduit_conductance’
stw_tp.settings[’conductance’] = ’throat.conduit_conductance’



Snwparr = []
Pcarr = []
Sarr=np.linspace(0,1,num=70)
Sat=[]
for pc in tqdm(np.unique(op1[’pore.invasion_pressure’])):
    # Update the occupancy of each phase at the current value of s
    water.update(op1.results(Pc=pc))
    Pcarr.append(pc)

    air[’pore.occupancy’] = ˜water[’pore.occupancy’]
    air[’throat.occupancy’] = ˜water[’throat.occupancy’]
    # Regenerate the physics’ to remove saturation effects from previous loops
    phys_air.regenerate_models()
    phys_water.regenerate_models()
    this_sat = 0
    this_sat += np.sum(pn[”pore.volume”][water[”pore.occupancy”] == 1])
    this_sat += np.sum(pn[”throat.volume”][water[”throat.occupancy”] == 1])
    Sat.append(this_sat)
    # Run all algorithms
    fda_tp.run()
    fda_ip.run()
```

```
    sta_tp.run()
    stw_tp.run()
    op.io.VTK.save(network = pn,
                   phases=[fda_tp,fda_ip,sta_tp,stw_tp,water],
                   filename=path+’any/op/water_op_’+str(len(Sat)))
    # Determine the flux into the top of the domain, in each phase
    deffa_tp.append(fda_tp.calc_effective_diffusivity(domain_length = im.shape[0]*voxel,
                                                      domain_area = im.shape[1]*im.shape[2]*voxel**2))
    deffa_ip.append(fda_ip.calc_effective_diffusivity(domain_length = im.shape[1]*voxel,
                                                      domain_area = im.shape[0]*im.shape[2]*voxel**2))
    ka_tp.append(sta_tp.calc_effective_permeability(domain_length = im.shape[0]*voxel,
                                                    domain_area = im.shape[1]*im.shape[2]*voxel**2))
    kw_tp.append(stw_tp.calc_effective_permeability(domain_length = im.shape[0]*voxel,
                                                    domain_area = im.shape[1]*im.shape[2]*voxel**2))



tot_vol = np.sum(pn[”pore.volume”]) + np.sum(pn[”throat.volume”])
Sat = np.asarray(Sat)
Sat /= tot_vol
Pcarr = np.asarray(Pcarr)

np.savetxt(path+”pnm_water_op.csv”, (Pcarr,Sat), delimiter = ’,’)
with open(path+’pnm_rel_OP.csv’, ’w’) as csv_file:
    csv_writer = csv.writer(csv_file, delimiter = ’,’)
    csv_writer.writerow(Sat)
    csv_writer.writerow(np.asarray(deffa_tp/deffa_tp[0]).flatten())
    csv_writer.writerow(np.asarray(deffa_ip/deffa_ip[0]).flatten())
    csv_writer.writerow(np.asarray(ka_tp/ka_tp[0]).flatten())
    csv_writer.writerow(np.asarray(kw_tp/kw_tp[-1]).flatten())
```

Appendix C

Code for MIP and PSD in OpenFCST

```
”””
Created on Thu Aug 1 11:05:36 2019

@author: yeop
”””

import pyfcst.ips.analysis as an
from pyfcst.ips.analysis.Intrusion import Intrusion
import pyfcst.ips.inout as io
import numpy as np
import matplotlib.pyplot as plt

################ load image stack
’’’
When the region of interest is not zero (not black), uncomment the commented line to invert the white region black
When sequence =”no”, 3D stack image file is read. To read the sequential images, set it ”yes”.
Ex)
sequential=”Slice_”
image = io.reader(temp_dir+sequential, sequence = ’yes’)
image = image.getarray()
’’’

path = ”/home/yeop/Documents/Article/image_gen/CL_recon/”
sample = ”stack1.tif”
image=io.reader(path+sample, sequence=”no”).getarray()

#image = np.invert(image) #Uncomment this when invert the white region to black.


################ Calculating PSD using sphere fitting algorithm
’’’
By setting material = 0, the module computes the distance map of black region. If it is set 255, the module will run
    ↪  with white region of the image. However, unlike the sphere fitting module can assign the region of interest
    ↪ , the intrusion module only works with the material = 0. When computing both PSD and MIP, set the
    ↪ material = 0 and invert the image color.
’’’
sphere = an.distanceSD(image,
                       label = ”Pore”,
                       voxelsize = [2e−9, 2e−9, 2e−9],
                       voxelunit = ”m”,
                       material = 0)
sphere.calcPSD(100)
#tag = ’Example_’
#sphere.plot_distn(datatype = ’cpsd’, label = tag)
#sphere.plot_distn(datatype = ’psd’, label = tag)
```

```
span = sphere.span
dxdr = sphere.psd/sum(sphere.psd)/((sphere.span[1])−(sphere.span[0]))
rdxdr = span∗dxdr


np.savetxt(path+’psd.csv’,(span, rdxdr, sphere.psd),delimiter=’,’)

################## Calculating virtual MIP
’’’
Intrusion module always works on the black region (intensity 0).
Check the color of the region of interest is black (or intensity 0).
’’’
obj = Intrusion(image,
                [3e−9, 3e−9, 3e−9],
                folder=path,
                mode=’MIP’,
                nsteps = 40,
                Contact_angle=140,
                Surface_tension=0.48,
                psd_type = ’single’)
poreIn=np.zeros(image.shape)
P,sat = obj.run(poreIn=poreIn, Pstart = 0, nsteps=40,mesh=”off”)

result = P, sat
np.savetxt(path+”fcst_mip.csv”, result, delimiter = ”,”)

# Calculating MIP based PSD
r = −(2∗0.48∗np.cos(140∗np.pi/180)/P)
r[0]=0
sat = np.asarray(sat)/100
dx = np.asarray([y−x for x, y in zip(sat, sat[1:])])
dr = np.asarray([x−y for x, y in zip(P, P[1:])])
rdxdr_mip= P[1:]∗dx/dr

plt.figure(0)
plt.plot(span,rdxdr)
plt.plot(r[1:], rdxdr_mip)
mip_psd = r[1:], rdxdr_mip
np.savetxt(path+”mip_psd.csv”, mip_psd, delimiter = ”,” )
```

Appendix D

Mesh generation in OpenFCST

```
#!/usr/bin/env python2
# −∗− coding: utf−8 −∗−
”””
Created on Wed Nov 14 13:15:06 2018

@author: yeop
”””



import pyfcst.mesh.PhaseGenerator as grid
from PIL import Image
import os
import numpy as np
import pyfcst.ips.inout as io
import scipy as sp

temp_dir = ’/home/yeop/Documents/Article/image_resizing/Ti/150cube_real/’+str(j+1)+’/’

##### Stack to image part
image = io.reader(temp_dir + ’stack1.tif’)
image = image.getarray()
image = np.array(image, dtype = np.float)
image = np.absolute(image)

os.makedirs(temp_dir+’/vtk/’ + ”Sample−1”+”/inputs/stack1/”)
for i in range(image.shape[2]):
    sp.misc.imsave(temp_dir +’vtk/’+ ”Sample−1”+”/inputs/stack1/” + ”Segmented−%03d.tif” %(i+1), image[:,:,
        ↪ i])

#Determine voxel sizes for all directions
imsize_origin = [300.0, 300.0, 300.0]
voxel_origin = 1.79e−4
coarsened = [150.0, 150.0, 150.0]
vox_ip = imsize_origin[0]/coarsened[0]∗voxel_origin
vox_tp = imsize_origin[2]/coarsened[2]∗voxel_origin
voxel = [vox_ip, vox_ip, vox_tp]
np.savetxt(temp_dir+”voxel_size.txt”, voxel, delimiter = ”,” )

#Defining variables to read in images; Omit this section if you already have a 3D array
# To generate VTK mesh

foldername = temp_dir+’/results/resized_images1/’
basename = ”resized_image”
num_images = int(coarsened[2])

output = temp_dir + ”sample−1.vtk”
name=foldername+basename+’%.3d’
extension=’.png’
```

```
tmp = []


#Looping over images and reading them
for k in range(num_images):
  filename=name%(k+1)+extension
  tmp.append(np.asarray(Image.open(filename),dtype=np.float))

#Generating the 3D numpy array
#changing all non-zero material ids to 20; assuming that pixel value of 0 was the one of interest
image=np.dstack(tmp)
image[image>0]=1
image[image==0]=255
image[image==1]=0
#Generating the VTK mesh using the Mesh module in PythonFCST

os.chdir(temp_dir)
o=grid(image,output,scale=voxel,limit=0)
o.write()
```

Appendix E

Python code for running PMSRT

```
”””
Created on Fri Oct 19 10:47:00 2018

@author: yeop
”””
from matplotlib import pyplot as pl
import pyfcst.ips.inout as io
import numpy as np
import scipy as sp
import subprocess
from numpy import genfromtxt
import pickle
import shutil
import sys,os,re

pl.rcParams[’figure.figsize’] = (7.5, 5)
pl.rcParams.update({’font.size’: 20})

temp_dir =’/home/yeop/Documents/Article/toray/600to200/’
Param_path = ’/home/yeop/Documents/Article/image_resizing/’
pmsrt_path = ’/home/yeop/pmsrt/Install/bin/’
files = ’/parameters.prm’
voxel1 = [1.79, 1.79, 1.79]

###### Stack to image part
image = io.reader(temp_dir + ’/stack1.tif’)
image = image.getarray()
image = np.array(image, dtype = np.uint8)
image[image>0] = 1
image[image==0] = 255
image[image==1] = 0

os.makedirs(temp_dir + ”/inputs/reference_images/stack1”)
for j in range(image.shape[2]):
    sp.misc.imsave(temp_dir + ”/inputs/reference_images/stack1/” + ”Slice_%03d.tiff” %(j+1), image[:,:,j].astype(
          ↪ np.uint8))

##### Read sequential images
im_path = temp_dir+’/inputs/reference_images/stack1/’
sequence=”Slice_0001.tiff”
image = io.reader(im_path+sequence,sequence = ’yes’)
image = image.getarray()


##### Run PMSRT#########
dest_path = temp_dir

shutil.copy(Param_path+files, dest_path)
```

```
args = (pmsrt_path + ”./Reconstruction”, dest_path + ”/parameters.prm”)
sample = dest_path
os.chdir(sample + ”/”)

pmsrt = subprocess.Popen(args, stdout=subprocess.PIPE)
pmsrt.wait()
print pmsrt.stdout.read()

skip_lines=0
first_string = ’Distance␣(Pixels)’

with open(sample + ’/results/PMSRT_Output.dat’) as myFile:
    for num, line in enumerate(myFile, 1):
        if first_string in line:
            skip_lines=num

myFile.close()
corr_functions = genfromtxt(sample + ’/results/PMSRT_Output.dat’, delimiter=’\t’,
                            skip_header=skip_lines, filling_values=’0.0’)
height=corr_functions.shape[0]
width=len(corr_functions[1])−1

corr_array=np.zeros((height,width))

for i in range(0,height):
    for j in range(0, width):
        corr_array[i][j]=corr_functions[i][j]

corr_pickle = open(sample + ’/results/corr’,’w’)
pickle.dump(corr_array,corr_pickle)
corr_pickle.close()
```