TORTUOSITY ANALYSIS OF POROUS POWDER COMPACTS

Assem Zharbossyn, B. Sc. in Chemistry

Submitted in fulfillment of the requirements for the degree of Master of Science in Chemical and Materials Engineering

School of Engineering and Digital Sciences
Department of Chemical and Materials Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Nur-Sultan, Kazakhstan, 010000

Supervisor: Dr. Boris Golman
Co-supervisor: Dr. Assiya Yermukhambetova

April 2020

Abstract

Advancement of consumer electronic devices and electric vehicle urges a need for batteries with higher power. Increasing the amount of active material is found to be ineffective as thicker electrodes may add limitations on transport characteristics. Thus, microstructure enhancement and analysis is a crucial step in the development of fluid transportation property of batteries. Therefore, the main objective of this work is to study the effect of porous structure parameters on the tortuosity of ternary powder compacts.

This thesis mainly reviews existing approaches in tortuosity evaluation of porous structures and presents results from DEM simulation and Voronoi graph assisted analysis of ternary powder compacts. Distribution of tortuosity factor was found for the ternary compacts applying standard Dijkstra's algorithm on the constructed Voronoi diagram. Comparison of tortuosity distribution curves of the ternary packing structures in terms of three different size ratios (r_{small}:r_{medium}:r_{large} 1 cm: 2 cm: 4 cm, 1 cm: 2cm: 6cm and 1 cm: 2 cm: 8cm) of particles and different volume fractions (f_{small}:f_{medium}:f_{large} 5%: 5%: 90%, 15%: 15%: 70%, 25%: 25%: 50%, 35%: 35%: 30% and 45%: 45%: 10%) of particles and different coordinate directions have been conducted. The results demonstrated a trend for tortuosity distribution peak to rise from fraction 5:5:90 to 45:45:10. Thus mixtures with higher fraction of small particles give narrower range of tortuosity factor values, while samples of higher proportion of large particles have a wider tortuosity distribution. This disposition becomes well defined for the samples, with the growth of large particles in size (1:2:6 and 1:2:8), resulting to more asymmetric and positively skewed tortuosity factor distribution. Another point is a lack of a distinct inclination of tortuosity factor distribution to change in a certain way with the alteration of coordinate axis direction. However, there are some deviations with respect to the z-axis, which can be associated either with difference in boundary conditions of walls orthogonal to z-direction from that of x- and y-directions or unevenness of particle arrangement as a consequence of the action of gravitational force during packing. The same tendency is observed in Voronoi cell edge lengths and face areas as in tortuosity distribution with respect to particle volume fraction, where the parameter distribution peaks reduce in height and extend towards larger values with the increase in large particle volume fraction. On the contrary, with respect to particle size ratio, the face areas are observed to show the opposite tendency of peaks to become

narrower and higher in fractions 15:15:70 and 25:25:50, and to shift to the in fraction 5:5:90, with the increases in size of a large particle. However, it can be claimed that appearance of higher values of the face areas and their increase in quantity can compensate the difference in tendency. Therefore, the relation between tortuosity and Voronoi parameters can be revealed, which states that with the increase in the distribution of Voronoi cell edges and face areas towards larger values, the distribution of tortuosity also increases towards large values and vice versa by enlarging the portion of smaller Voronoi cell edges and faces tortuosity factor reaches smaller and more uniform values. Due to inverse relation between the diffusivity and tortuosity, smaller Voronoi cell edges and faces contributes to a better diffusivity.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Dr. Boris Golman, who provided all the opportunities to implement this research and my co-supervisor Dr. Assiya Yermukhambetova. I am thankful for all the guidance, support, patience and consultations they delivered throughout the research period. I am also thankful to our team members Zhazira and Aidana, who helped me and become friends.

My sincere appreciation to my parents Bekdilla and Sveta, who encouraged my interest in science; to my sister Aktoty, who never stopped believing in me.

Table of contents

List of Abbreviations

DEM – Discrete Element Method

LIB – lithium-ion battery

FIB-SEM – focused ion beam and scanning electron microscope

LBM – lattice Boltzmann method

FMM – fast marching method

RWPT – random-walk particle-tracking

STEM – scanning transmission electron microscopy

2D – two-dimensional

3D – three-dimensional

List of Tables

List of Figures

Chapter 1 – Introduction

1.1 General

It is possible to convert one form of energy to another and to store it in different forms. One of the popular electric energy-storing forms is battery – an electrochemical device, which allows portable electricity concept. This has created an opportunity not only for consumer electronic devices but also for electric vehicle revolution fully or partially working on batteries. Due to sharply continental climate in Kazakhstan, the severe climatic conditions in winter greatly affect the performance of the battery and lifetime requiring a part of the battery energy for self-heating [1]. In this respect, a battery with high energy density and high power density is required. One way of achieving high energy density with existing commercial battery technologies is to increase the amount of active material in a single cell and thus, there is a question of proper packing structure of electrodes as thicker electrodes may add limitations on transport characteristics.

Determination of quantitative dependence between effective fluid transport properties and morphology of porous powder compacts is one of the important issues in catalysis, adsorption, separation, and energy applications [2]. Since most industrial applications of such compacts are based on efficient fluid transport through compact, regardless of their functionality, targeted compact characteristics optimization gives an insight into relevant mass transport properties (and possible transport constraints). It is required to gather adequate knowledge on the outcomes of conditions of material preparation for the arising morphology of three-dimensional compact and the transport properties associated with it. Thus, the main **objective** of the work is to study the effect of porous structure on the transport properties of powder compacts via obtaining tortuosity.

1.2 Aims and objectives

This study aims

(1) to simulate the compaction of spherical particles of various sizes using the Discrete Element Method (DEM);

(2) to characterize quantitatively and analyze the packing structure of the obtained porous powder compacts in detail using methods such as Voronoi tessellation; and

(3) to investigate the effect of the obtained packing structure on fluid transport properties through compact.

1.3 Thesis layout

This thesis is organized as follows: Chapter 2 provides general introduction to the concept of tortuosity and to the discrete element method (DEM). Moreover, the description of two main routes of tortuosity calculation is reviewed in this chapter. In Chapter 3, the methodologies are developed for the formation of powder compact using DEM, analysis of porous structure of powder compacts applying Vorornoi tessellation and simulation of transport properties of fluid through the compact. Chapter 4 reports the results of conducted simulations and related analysis. Finally, the conclusions are drawn and the recommendations for future research work are provided in Chapter 5.

1.4 Contribution

Major contributions made by the author:

- Generation of powder compacts of polydisperse spherical particles with different compositions using open-source software package LIGGGHTS.
- Construction Voronoi tessellation of theobtained powder compacts on Voro++.
- Tortuosity factor calculation on Matlab and obtaining its distribution.
- Study the relationships between the compact packing structure and tortuosity.

Chapter 2 – Literature Review

2.1 Porous structure and lithium-ion batteries

Numerous up-to-date studies on battery technologies emphasized the lithium-ion batteries (LIB) as the dominant one in automotive applications primarily due to their comparatively high energy density and high power density. Nevertheless, it is well known that when developing a new cell, it is necessary to achieve the optimal trade-off to reach the target energy and power indicators, which are announced by many vehicle manufacturers and national research bodies. The quantification of the trade-off have been attempted by recent researches, both through experimental evaluation and numerical modeling. The main trend in the presented results is that an increase in power density is possible only by reducing the energy density. The power density improvement can be reached either through the replacement of an active material portion with conductive fillers and use of large pores for a better ion transportation, or by using thinner electrodes [3].

A research on geometry optimization of LIB using finite element method (FEM) based simulation have been conducted considering conventional as well as different unconventional geometries such as ring, gear, horseshoe, antenna and spiral batteries [4].Mitchell et. al attempted to improve LIB performance applying computational topology optimization techniques for a silicon anode. In order to diminish the anode swelling and enhance the electron conductance, they considered structure and conduction design of the anode [5].

Regardless of the certain choice of the chemical composition of the battery, cell performance can be improved by varying critical design parameters during manufacturing process, including porosity, particle size, electrode thickness, electrode surface area and current collector geometry and dimensions [6]. It was shown that transport properties, electronic and ionic conductivity have a considerable influence on the performance of LIB. Applications allowing a slow discharge rate can use the choice of thicker electrodes with low porosity, while thinner highly porous electrodes are more suitable for applications with short discharge time and large power.

However, a huge number of experiments should be done in the interest of establishing all the specific relationships. As long as experimental research involves high

costs of processes with respect to resources, raw materials and time, numerical modeling of LIB have recently been gaining a lot of attention as an alternative approach.

2.2 Porous structure and transport properties

The relevant compact transport properties comprise hydrodynamic dispersion and effective diffusion coefficients, which respectively indicate to mass transport mostly based on advection in large pores (macropores) and diffusion-dominated transport in small pores (micropores and/or mesopores). The various size pore spaces are necessary for fluid flow and, as a rule, subject to a hierarchical structure [7,8]. For instance, macropores ($d_{pore} > 50$ nm, as stated in IUPAC nomenclature) form interparticulate and interskeleton empty space of, respectively, particle packed beds and monoliths (utilized as separators, adsorbers and reactors), whereas porous particles and monolith skeleton can contain micropores ($d_{pore} < 2$ nm) and/or mesopores ($2 \text{ nm} \leq d_{pore} \leq 50$ nm) (Figure 2.1) [2]. Common mesoporous materials comprise some of alumina and silica, and primarily carbon applied in devices for energy storage, with equally sized mesopores. Microporous materials have their frequent use in laboratory conditions for exchange of gases without contaminant and many other uses such as microporous adhesive tape in medicine, highly efficient insulation material, etc.

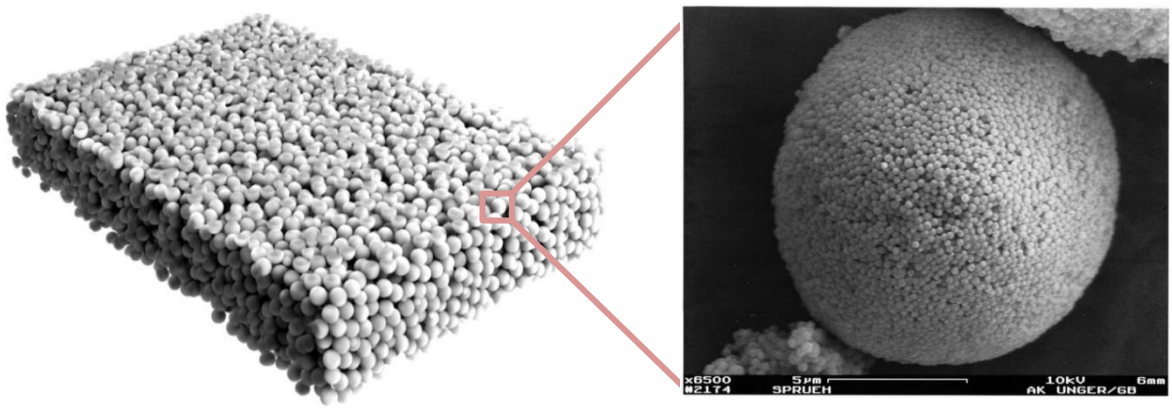


Figure 2.1: Scanning electron micrograph of: left: 1 µm spherical silica particles; and right: a nanoporous bead with 1 µm size constructed from spherical nanoparticles obtained by the spray drying technique [2].

2.3 Diffusion coefficient calculation methods

This study is focused on mass transport through micro- and mesopores and thus on effective diffusion coefficient, which can be determined by random walk [9] simulation or through tortuosity calculations [9-14]. Many researchers have been interested in tortuosity evaluation methods. Kishimoto et al. (2009) [10] evaluated transport properties such as tortuosity of SOFC anode applying random walk method to 3D data obtained from a focused ion beam and scanning electron microscope, FIB-SEM. They revealed that larger surface-to-volume ratio of Ni phase is required for microstructure optimization. Tortuosity factor evaluation studies of Iwai et al. (2010) [11] compared two different methods of tortuosity factor evaluation, random walk and lattice Boltzmann method (LBM), showing differences of less than 3% between them. Jogensen et al. (2011) [12] used the fast marching method (FMM) to compute arrival time maps for tortuosity calculation. They compared in detail tortuosity distributions between several samples, elaborated and concluded that the geometrical tortuosity evaluation are made without any phase assumptions. Tjaden et al. (2016) [13] also used FMM as image based method together with simulation based methodology and experiments in diffusion cell for their reliability analysis. According to their results, image and simulation based methods derived tortuosity can be lower than that of experimental one by factor of two. Semeykina and others (2018) used tortuosity to calculate the diffusion coefficient to determine the optimal catalyst texture in the conversion of macromolecules [14]. They used Voronoi diagram for the simulation of the system forming void space between the grains and applied Dijkstra algorithm for tortuosity calculations. However, there is no adequately addressed relationship between morphology parameters and transport properties such as effective diffusivity. Thus, the present study is aimed to disclose if there is some relationship between Voronoi parameters and effective diffusivity which will provide opportunity to anticipate optimal decisions for parameters such as particle shape, size distribution, and composition.

Further, this literature review will cover fundamental concepts of diffusion and two main studies on two methods of effective diffusion coefficients calculation: Hlushkou et al. (2017) [9] and Semeykina et al. (2018) [14].

2.4 Diffusion and random-walk particle-tracking (RWPT)

Diffusion is the process of redistribution of elements in a system under the influence of the thermal motion of atoms and molecules. In fact, diffusion is not a process proceeding under the influence of any special force but is the result of the random motion of atoms, i.e., a statistical problem. In a more general consideration of the phenomenon, the atomic nature of the diffusing particles can be neglected and their quantity determined as the concentration C, that is, the number of particles per unit volume [n/L^3, i. e. $1/m^3$]. The combination of the first Fick law with the continuity equation is the second Fick law:

$$\frac{\partial \varphi}{\partial t} = D\nabla^2\varphi, \quad (1)$$

where φ is the concentration of diffusing particles, t is the time, and D is the diffusion coefficient.

Using the approach to diffusion as a process of random walks, it is possible from a practical and theoretical point of view to obtain an extremely useful equation, often called the parabolic law developed by A. Einstein in 1905 [15], which leads to the following equation:

$$\overline{X^2} = 6Dt \text{ or } \bar{X} = \sqrt{6Dt}, \quad (2)$$

where $\bar{X}$ is the average distance that the diffusing particle travels during the experiment, and 6 is for one-dimensional diffusion in cubic crystals.

The value $(Dt)^{1/2}$ is often called the characteristic diffusion path; that is, the average distance the diffusing particle travels during the experiment [16]. Hlushkou et al. (2017) used this equation to develop an algorithm for the long-term diffusion coefficient D_{eff} calculation in amorphous, mesoporous silica by random-walk particle-tracking (RWPT) technique in a cubic domain reconstructed by scanning transmission electron microscopy (STEM) tomography (Figure 2.2) [9]. The algorithm is briefly described by a flowchart shown in Figure 2.3: first, they randomly distributed $N = 5 \cdot 10^6$ number of tracers, which is roughly equal to the number of voxels of the void part of the domain, across the void space. During every δt elementary time steps of a random walk when displacement does not exceed $\frac{\Delta h}{10}$ ($\Delta h = 0.47\ nm$ is the spatial resolution of the reconstruction), each tracer replacement was computed by Gaussian distribution with a mean equal to zero and standard deviation equal to $\sqrt{6D_m\delta t}$, where D_m is the free space molecular diffusion

coefficient. Thus, the equation reflecting the tracer random walk can be represented as the following discrete form of the stochastic differential equation:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \boldsymbol{\alpha}\sqrt{6D_m\delta t}, \quad (3)$$

where $\mathbf{r}(t)$ refers to a tracer position at time t, δt is the elementary time step in a random walk, and $\boldsymbol{\alpha}$ is the randomly-oriented vector.

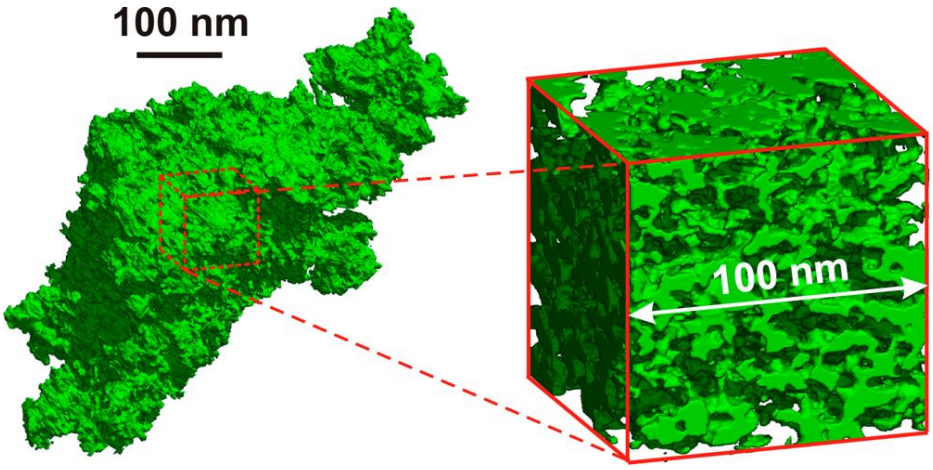


Figure 2.2: Left: reconstructed by STEM crumb of mesoporous silica with an irregular shape; Right: the cubic domain cut for application in the simulation of the finite-size tracer particles diffusion [9].

After monitoring positions in every time step, the time-dependent diffusion coefficients D(t) were computed according to each tracer displacements:

$$D(t) = \frac{1}{6N}\frac{d}{dt}\sum_{i=1}^{N}[\Delta r_i(t) - \langle\Delta r(t)\rangle]^2. \quad (4)$$

Here, $\Delta r_i(t)$ and $\langle\Delta r(t)\rangle$ are the ith tracer and tracer ensemble displacement during time t, respectively.

Nine simulations were performed for the diffusion of finite-sized particles differing in radii in the similar porous media by decreasing available void space for the centers of the finite-sized particles. In other words, they reduced the pore space by the value of finite-size tracer diameter and point-like tracers were used as the centers of finite-sized particles in each changed domains. After all, due to long-time asymptotes of the curves of the time-dependent diffusion coefficients $D(t)$, the effective diffusion coefficients D_{eff} were defined for each case (Figure 2.4).

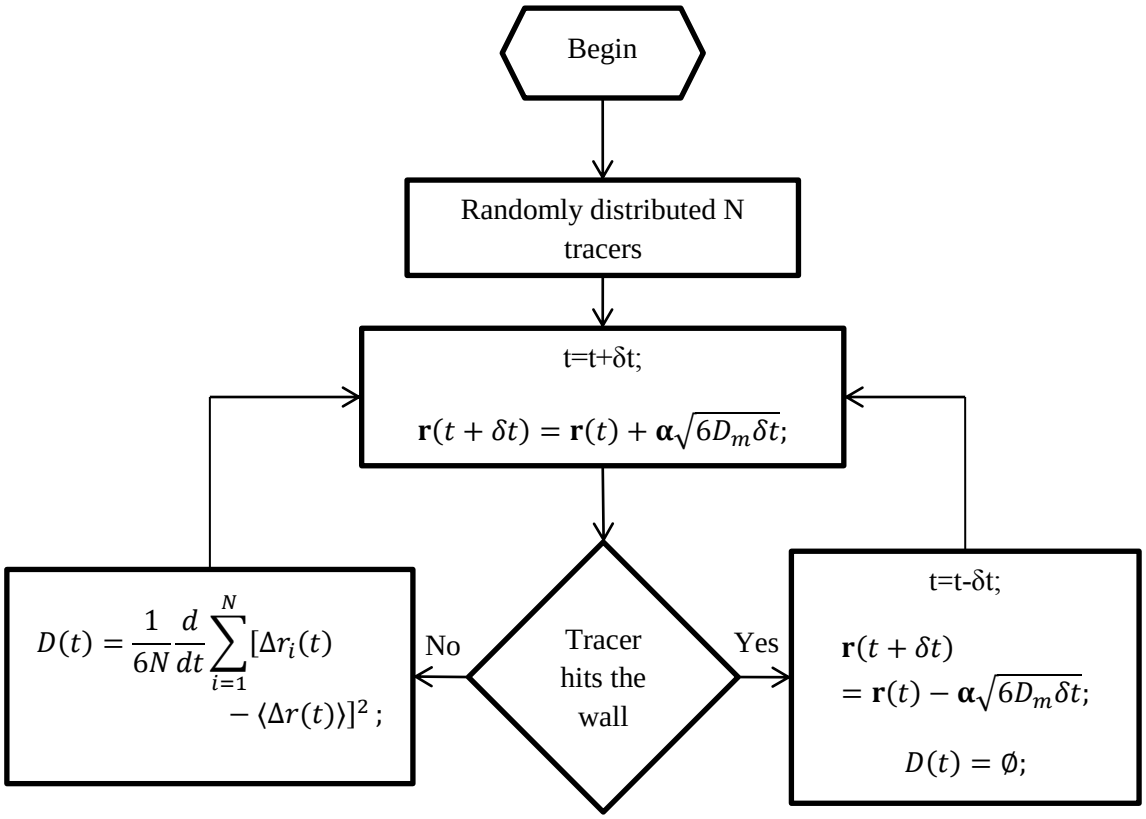


Figure 2.3: Flowchart constructed for the algorithm of long-term diffusion coefficient calculation by RWPT technique according to Hlushkou et al., (2017) [9].

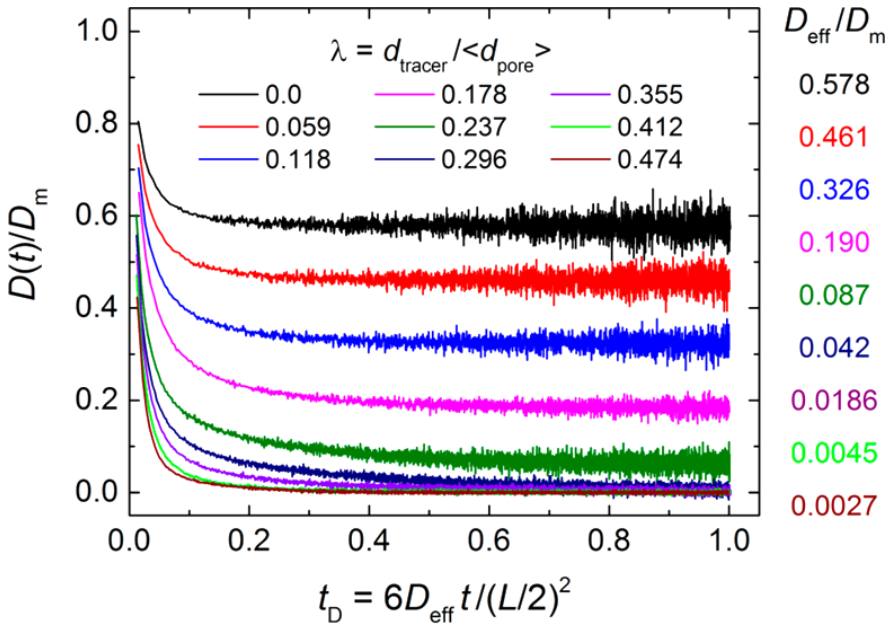


Figure 2.4: Dependence curves of time-dependent diffusion coefficients D(t) of the nine different finite-sized traces in the reconstructed domain, processed through normalization by the free space molecular diffusion coefficient Dm, on the dimensionless diffusion time $t_D = 6D_{eff}t/(L/2)^2$ [9].

2.5 Tortuosity calculation methods

Since the present study is focused on mass transport through micro- and mesopores and thus on effective diffusion coefficient, it can also be expressed as follows:

$$D_{eff} = D_{bulk} \cdot \frac{\varepsilon}{\tau}, \tag{5}$$

where D_{eff} is the effective diffusion coefficient, D_{bulk} is the combination of Knudsen and molecular diffusivity in a straight pore, ε is the respective network porosity and τ is the tortuosity. Compared to liquid-phase diffusion coefficient (10^{-6} – 10^{-5} m^2/s) in electrolytes, the effective diffusion coefficient in a porous medium is several magnitude lower (10^{-10} – 10^{-15} m^2/s order of magnitude) [17,18].

When diffusion models are used for evaluation and quantification the transport of fluid species through porous compacts, microstructural compact characteristics such as porosity, tortuosity, and average radius of porous space are very influential for procedures of calculation. Many studies focus on estimating the tortuosity τ, which is geometrically determined as the ratio of the shortest path through the porous compact to the Euclidean length between the endpoints of this path herewith $\tau > 1$. Porous functional layer tortuosity, as a microstructural characteristic, is considered as a constant parameter independent of the diffusing fluid or operating temperature. To extract the porous sample tortuosity, there are several developed methods that are significantly different from each other. This generates the question of most suitable method for obtaining a representative tortuosity to predict how porous compacts affect the transfer of fluid species.

One of these techniques is the characterization of a porous sample microstructure from three-dimensional images which can be obtained using X-ray computed tomography (CT) [19] or focused ion beam scanning electron microscopy (FIB-SEM) slice and view tomography [20]. Then, the tortuosity can be obtained using image-based tortuosity algorithms such as the fast marching method [12], the Lattice-Boltzmann method [20], the random walk method [21 – 23], or the pore centroid method [24,25].

As an alternative, directly modeled transport processes throughout the reconstructed data sets within the microstructure can be used to extract effective transport parameters and, eventually, tortuosity applying computational fluid dynamics methods [25 – 28].

The direct measurement of the diffusion fluxes throughout a porous membrane is another tortuosity assessment approach, in which diffusion cells are used, such as the cells of Graham and Wicke Kallenbach [29]. Then tortuosity is estimated by using a suitable

model of diffusion, such as the Stefan – Maxwell model, the dusty gas model or the Fick law [30].

2.6 Calculation of tortuosity applying Voronoi diagram

Semeykina and others (2018) used tortuosity to calculate the diffusion coefficient to determine the optimal catalyst texture in the conversion of macromolecules [14]. According to Semeykina et al., (2018), tortuosity can be calculated via Voronoi diagram. For the simulation of the system forming void space between the grains, they used the **Voronoi diagram** (Figure 2.5). The diagram consists of a set of polyhedron cells C_e enclosing all points of coordinate space closer to p_e than to any other p_f and having the following properties:

(1) every two cells C_e and C_f share one face C_{ef};

(2) every three cells C_e; C_f and C_g share one edge C_{efg};

(3) every four cells C_e; C_f; C_g and C_h share one vertex C_{efgh} [14].

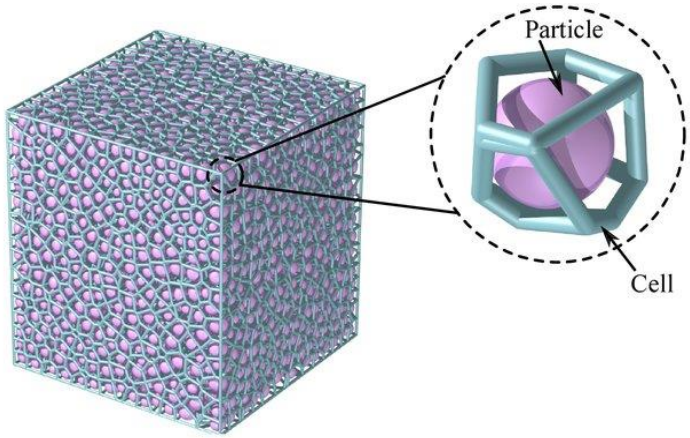


Figure 2.5 Voronoi tessellation for packings with arbitrarily arranged monodisperse spheres [31].

Obtained one-dimensional Voronoi diagram is composed of V set of vertexes and E set of edges. The vertexes (points) are pores located between grains and edges are tunnels linking pores. M number of pairs of q_j and s_j vertices was considered, q_i lying on one of the domain faces and s_i lying on the opposite face of the domain. Using Dijkstra algorithm [32] the shortest path $dist(q_j, s_j)$ for every pair of q_j and s_j was calculated, to find the tortuosity τ_{q_j,s_j} for the every pair and the mean tortuosity $\tilde{\tau}$ by:

$$\tilde{\tau} = \frac{1}{M}\sum_{j=1}^{M}\tau_{q_j,s_j} = \frac{1}{M}\sum_{j=1}^{M}\frac{dist(q_j,s_j)}{l}, \quad (6)$$

where l is the Euclidean distance.

Then, the tortuosity is set to Eq. (5), to find the effective diffusivity. The flowchart of the simulation process is shown in Figure 2.6.

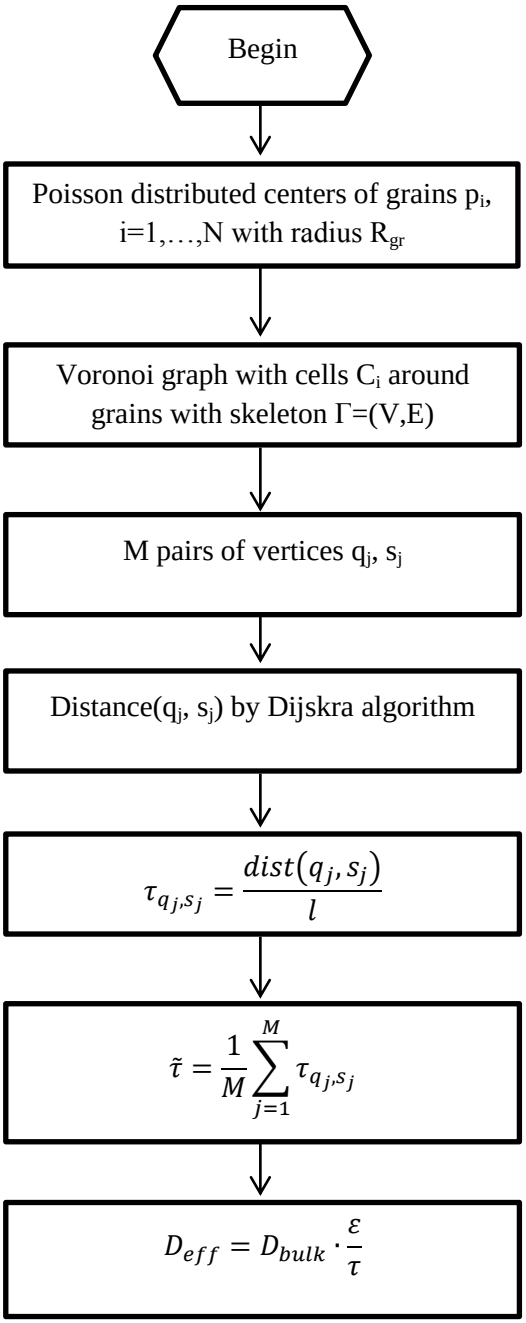


Figure 2.6: Flowchart constructed for the algorithm of effective diffusion coefficient calculation through tortuosity determination according to Semeykina et al., (2018) [14].

The application of the Voronoi diagram for tortuosity calculation [14] is more time-efficient, compared to the algorithm based on the RWPT method [9]. After comparing the two methods, we will use the method of applying the Voronoi diagram [14] in view of its efficiency and lower time consumption. This work will find the dependence of the transport properties of powder compacts on their porous structure by calculating tortuosity factor for different packing structures.

2.7 The Discrete Element Method (DEM)

The Discrete Element Method (DEM), proposed by Cundall and Strack [33] as a powerful and versatile discontinuous numerical approach, made possible to model the motion of a large number of particles of granular systems [34 – 36], as well as the micromechanical understanding of its behavior.

By means of the DEM, a material is discretized to simple-shape rigid elements while neighboring elements interact with each other in accordance with the laws of interaction at the points of contact. Although the rigid elements can be of different shapes, most often they have a round or spherical shapes due to the contact detection algorithm speed and simplicity. There are three main computational stages in the analysis procedure: (1) evaluation of internal force (calculation of contact forces); (2) integration of motion equations (computation of the element displacements); (3) contact detection (identification of new contacts and removal of broken contacts). The DEM analysis considers the element interaction as a dynamic process with alternation of the Newton's second law application with the force-displacement estimation law at the points of contact. Newton's second law identifies an element acceleration that arises as a result of the action of forces on it, comprising internal forces prescribed to the inter-element contacts, external forces arising from the boundary-element contacts, and gravitational forces. Then velocity and displacement are derived by the integration of the acceleration. New contact forces are found from known displacements by the force-displacement law. The central difference method is used to integrate the equations of motion in time. In [33] this process is described in detail.

Modeling particulate materials by DEM

It is common knowledge that granular disconnected materials such as sand and powders have different mechanical behavior from other materials in important aspects, and that this is because of the dispersed nature of the medium. Thus, the DEM is increasingly

being accepted and extensively used for modeling the mechanical behavior of particulate materials and their flow. Over the past few years, the scope of application of industrial materials has expanded with the modeling of the transportation of materials using conveyor belts, dragline excavators and ball mills. To a large extent, separate research activities are underway for powder and material processing modeling.

From the point of view of computational convenience in DEM, the elements of a 2D circular disk and 3D spherical shapes are ideal choices.

2.8 Conclusion

The purpose of the literature review was to understand the science behind the relation between porous structure and transport properties and look through the existing methodologies of diffusivity and tortuosity factor computation for porous structures. Two main computational methods, one of which uses random-walk particle tracking (RWPT) [9], and the second Voronoi diagram, were disclosed in more detail. After the two methods compared, the method applying the Voronoi diagram [14] is chosen for this study, in consideration of its cost efficiency and lower time consumption. This method will be employed to find the relation between the transport properties of several powder compacts and their porous structure by calculating tortuosity factor for different packing structures.

The review also covers general introduction to the concept of the discrete element method (DEM) which is used for the modeling of packing structure in this research.

Chapter 3 – Methodology

The assessment of the tortuosity of powder compacts included three main steps (Figure 3.1): (1) generation of powder compacts of polydisperse spherical particles with different composition using the DEM; (2) construction Voronoi tessellation of the powder compacts; (3) tortuosity calculation through Dijkstra's algoritm using data obtained from Voronoi tessellation. After that, the effect of Voronoi parameters (Voronoi cell volumes) on the obtained tortuosity was studied. Finally, the relationships between the compact packing structure and transport properties (tortuosity) were developed.

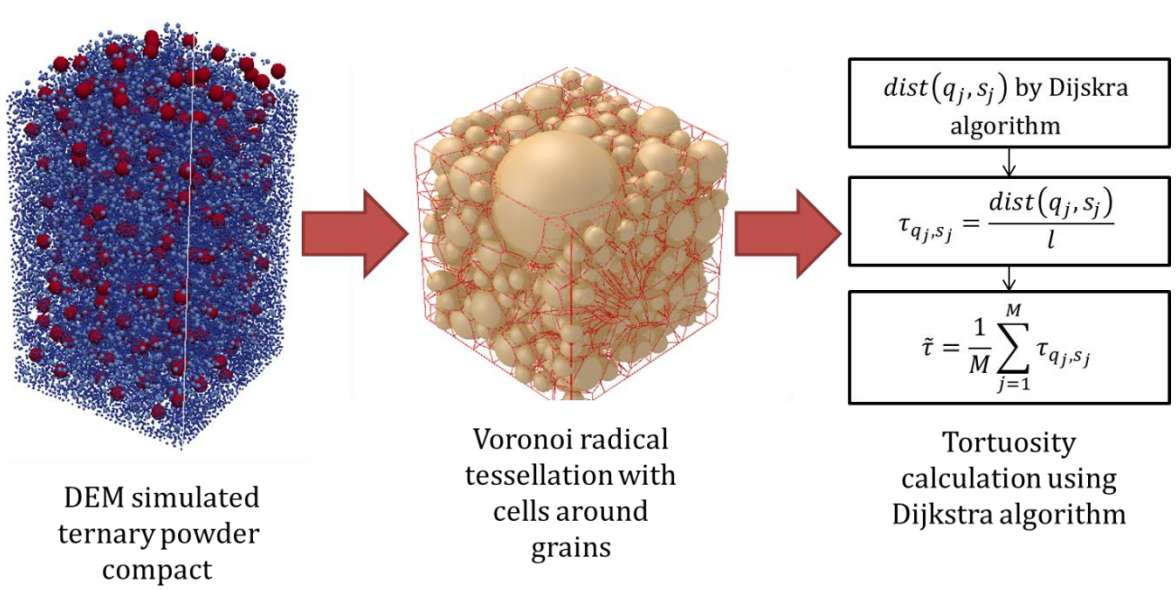


Figure 3.1: Flowchart of the algorithm of tortuosity coefficient calculation.

3.1 DEM simulation

In this study fifteen electrode structures of ternary powder compacts with different fractions were generated using open-source software package LIGGGHTS based on the DEM [37]. The DEM simulation scripts are given in Appendix A. A 3D box with size 0.8 m × 0.8 m × 3 m and with boundary conditions being periodic (pp) alongside the walls orthogonal to x- and y-directions and fixed (ff) through the boundaries orthogonal to z-axis was used for carrying out the simulation (Table 3.1). In the course of particle compaction

the wall effect is reduced owing to its periodic boundaries. The packing structure was generated under gravity.

In the beginning, ternary particle mixture of designated composition was arbitrarily distributed in the 3D domain. Then, during the simulation, gravitational force was applied for 7,000,000 time steps to reach a compact packing. The mechanical characteristics are given in Table 3.2. The ternary mixtures modeled were of three size ratios 1:2:4, 1:2:6 and 1:2:8 and with seven different volume fractions each, as shown in Table 3.3. Since the sizes of the particles have been set to be several centimeters, much larger than in real electrodes, Young's modulus was chosen 3 magnitude lower than that of real graphite electrode as in [38] and time step also was 2 magnitude lower than it necessary for finer particles. Such kind of calibrations are made in view of their time effectiveness.

Table 3.1: Particle mechanical properties and DEM parameters.

DEM parameters	
Box size, m	0.8×0.8×3
Boundary conditions x-y-z	pp-pp-ff
Number of particles, N	5000-70000
Gravity, g (m/s^2)	9.81
Time step (s)	$1*10^{-5}$

Table 3.2: Particle mechanical properties and DEM parameters.

Mechanical properties	
Particle density, ρ (kg/m^3)	2500
Poisson ratio, σ	0.45
Young's modulus, Y (N/m^2)	$5*10^6$

Table 3.3: Ternary mixture properties for DEM simulation.

Particle radii (cm) 1:2:4	#	Vol. frac., %			Particle radii (cm) 1:2:6	#	Vol. frac., %			Particle radii (cm) 1:2:8	#	Vol. frac., %		
		s	m	l			s	m	l			s	m	l
	1	5	5	90		**6**	5	5	90		**11**	5	5	90
	2	15	15	70		**7**	15	15	70		**12**	15	15	70
	3	25	25	50		**8**	25	25	50		**13**	25	25	50
	4	35	35	30		**9**	35	35	30		**14**	35	35	30
	5	45	45	10		**10**	45	45	10		**15**	45	45	10

3.2 Radical Voronoi tesselation

The obtained samples were cut from the 3D box to smaller cubes with wall size of 0.2 m for further analysis, in order to remove possibly irregularities on the top part. The volume of particles in the box was calculated by taking into account the particle-particle and particle-box wall overlaps.

The Radical Voronoi tessellation for packed structures was created using the open-source software package Voro++. The script of the 'chem.cc' file needed for the simulation is demonstrated in Appendix B. By Voronoi tessellation, the porous space of the compacts is divided into cells so that there was one particle per each polyhedron cell. Apart from dividing the compact into Voronoi cells, the software computes cell parameters such as number of faces, cell volume, total surface area, etc. [39]. The compact structure and its Voronoi diagram was visualized using the open-source software POV-Ray.

3.3 Implementation of Matlab calculation of tortuosity using Dijkstra's algorithm

Tortuosity calculations were conducted on Matlab software for every case by altering particle ratios, volume fractions and coordinate directions. One sample of simulation scripts for sample with particle size ratio 1:2:4, volume fraction 5:5:90 and x-direction is given in Appendix C. The file works only in the presence of 'dijkstra.m', the file with the Dijkstra algorithm (Appendix D).

Tortuosity was calculated for all possible M pairs of q_i and s_i vertices lying on the domain faces opposite to each other. Using Dijkstra algorithm [32] the shortest path

$dist(q_j, s_j)$ for every pair of q_i and s_i was calculated, to find the tortuosity τ_{q_j,s_j} for the every pair by:

$$\tau_{q_j,s_j} = \frac{dist(q_j, s_j)}{l}$$

where l is the Euclidean distance. Then data on the distribution of this M number of tortuosity was obtained.

As the lengths of the Voronoi cell edges are not provided by Voro++ outcomes, they were also calculated in Matlab from their coordinates and the information if there is an edge between two vertices (Appendix E).

3.4 Description of Dijkstra's algorithm

The shortest paths between pairs of the Voronoi graph vertices located on the opposite boundaries of the box was found through Dijkstra's algorithm. The algorithm was formulated by computer scientist Edsger Wybe Dijkstra in 1956, three years before it was published [32]. The steps of finding the shortest path are summarized:

1. Marking of all nodes as unvisited and creation of a set of unvisited nodes.
2. Assigning a preliminary distance value to every node: setting it to infinity for all nodes except the starting node for which it should be equal to zero. The starting node is marked as current [40].
3. Calculating the preliminary distances through the current node to all of its unvisited neighbors. The latest preliminary distance is compared to the already assigned value and then the smaller one is assigned.
4. When considering all of the unvisited neighbors is done for the current node, the node is marked as visited and deleted from the set of unvisited nodes. There will no more checking of a visited node.
5. The algorithm is finished if the finish node has a mark visited or if the nodes from the unvisited set have the smallest preliminary distance equal to infinity (occurs when the starting node and the rest unvisited nodes are not connected). If not, the unvisited node with the smallest preliminary distance is set as the next current node and the step 3 is repeated back.

3.5 Average values

Average values are used to demonstrate the central trend in the population of values of tortuosity.

The mode (τ_m) is the most frequently existent value in tortuosity distribution. It is the maximum frequency value.

The median (τ_{50}) is the value of tortuosity in which the distribution is divided into two parts with equal quantity of tortuosity value. It is the tortuosity corresponding to the 50% on the cumulative distribution curve.

The mean ($\bar{\tau}$) is the sum of tortuosity values divided by the number values:

$$\bar{\tau} = \frac{\sum_i^n \tau_i}{n}$$

τ_{10} is the tortuosity corresponding to the 10% on the cumulative distribution curve.

τ_{90} is the tortuosity corresponding to the 90% on the cumulative distribution curve.

$$span = \frac{\tau_{90} - \tau_{10}}{\tau_{50}}$$

3.6 Summary

To sum up, the simulations were conducted in three steps. First, DEM simulations of ternary powder compacts with different size ratios of spherical particles (r_{small}:r_{medium}:r_{large} 1:2:4, 1:2:6 and 1:2:8) along with different volume fractions of particles (f_{small}:f_{medium}:f_{large} 5:5:90, 15:15:70, 25:25:50, 35:35:30 and 45:45:10) have been conducted. Second, radical Voronoi tessellation of the obtained packing structures has been constructed. Third, tortuosity distribution through modeled powder compacts was calculated applying Dijkstra's algorithm to Voronoi graph.

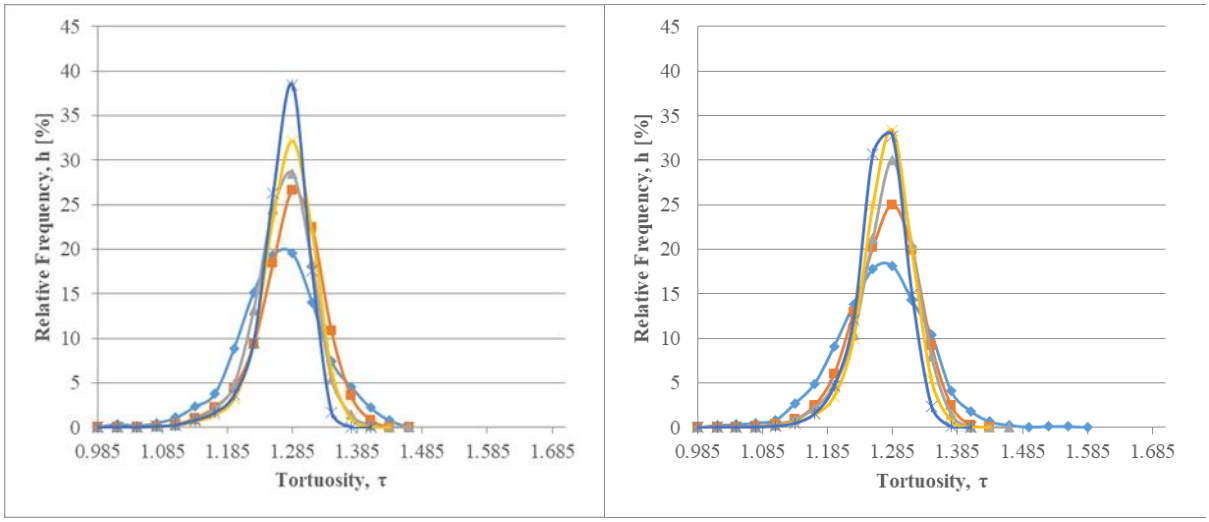
Chapter 4 – Results & Discussion

The comparisons of samples with different particle fractions in term of the relative frequency of tortuosity measured in x, y and z directions on compacts with particle size ratios 1:2:4, 1:2:6 and 1:2:8 are shown in Figures 4.1, 4.2 and 4.3, respectively. The values of mean tortuosity ($\bar{\tau}$), mode (τ_m) of tortuosity distribution and the median (τ_{50}) sizes and span of cumulative undersize distributions for the same ternary compacts are summarized in Figures 4.2, 4.3, 4.4 and 4.5, respectively. Figure 4.4 and 4.5 represent the relative frequency of, respectively, edge lengths and areas per face of Voronoi polyhedrons for these compacts.

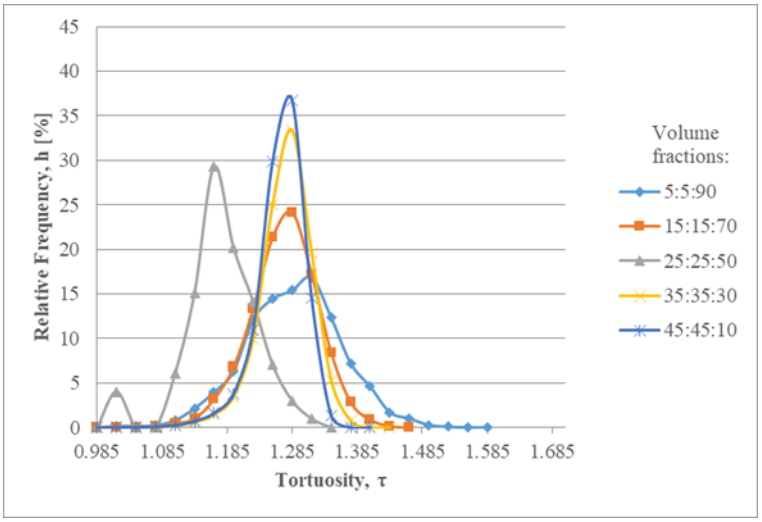
4.1 Relation between tortuosity and parameters of ternary mixture.

Figure 4.1 demonstrates that the peaks of tortuosity relative frequency in all directions decrease from around 35% to approximately 17% by the increase of the large particle volume fraction from 10% to 90%. For the samples with greater amount of large particles, the distribution lasts for higher values (1.555 for 90%), while the distribution for compacts with fewer large particles ends with lower tortuosity (1.375 for 10%).

As can be seen from Figure 4.2, the values of mean tortuosity for all samples are near to 1.27 in all directions, except the sample 25:25:50 in z-direction, which is 1.1764. According to the Figure 4.3, most of the mode tortuosity values are equal to 1.285, the most different being equal to 1.165, again for the sample 25:25:50 in z-direction. The median tortuosity (Figure 4.4) has the same pattern, most of them being around 1.26 and for the sample 25:25:50 in z-direction being equal to 1.16. The span (Figure 4.5) falls from approximately 0.127 to barely 0.069 by the reduction of large particle fraction in volume from 90% to 10%. The anomaly of sample 25:25:50 in z direction for the case 1:2:4 is probably attributed to an uneven arrangement of particles in the direction of gravity force during packing. From figures (Figure 4.6 and 4.7) visualized by POV-Ray, it can be seen that small particles are mostly arranged in the bottom part of the box, while large particles are mostly in the top part. The segregation of particles during packing under gravity can take place because the small particles have more chances to penetrate through voidages to the bottom than the large particles.



(a) (b)



(c)

Figure 4.1: Relative frequency of tortuosity of ternary compacts with particle size ratios 1:2:4 and different volume fractions measured in: (a) x-, (b) y-, and (c) z-directions.

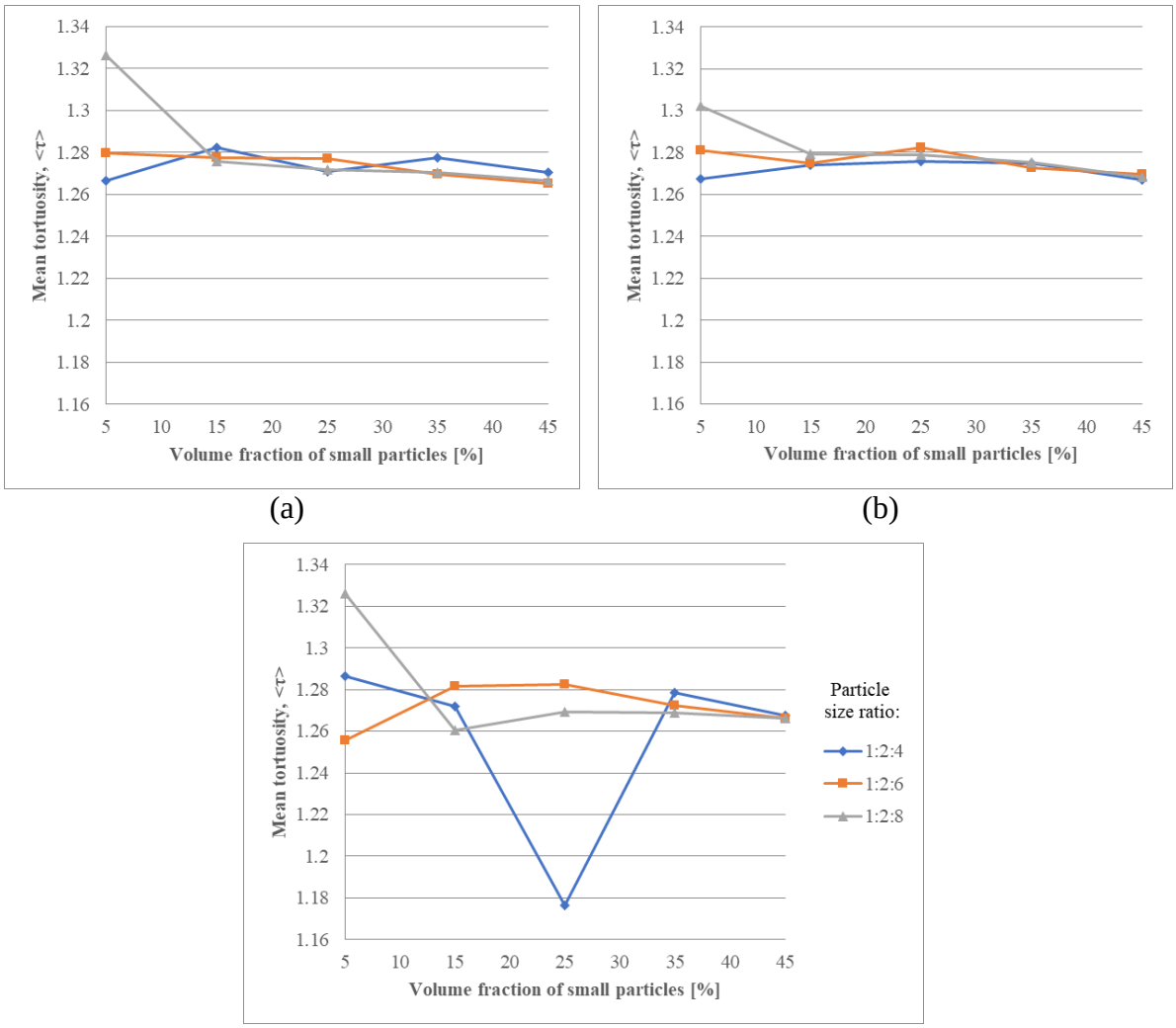


Figure 4.2: Mean tortuosity vs Volume fraction of small particles in: (a) x-, (b) y-, and (c) z-directions.

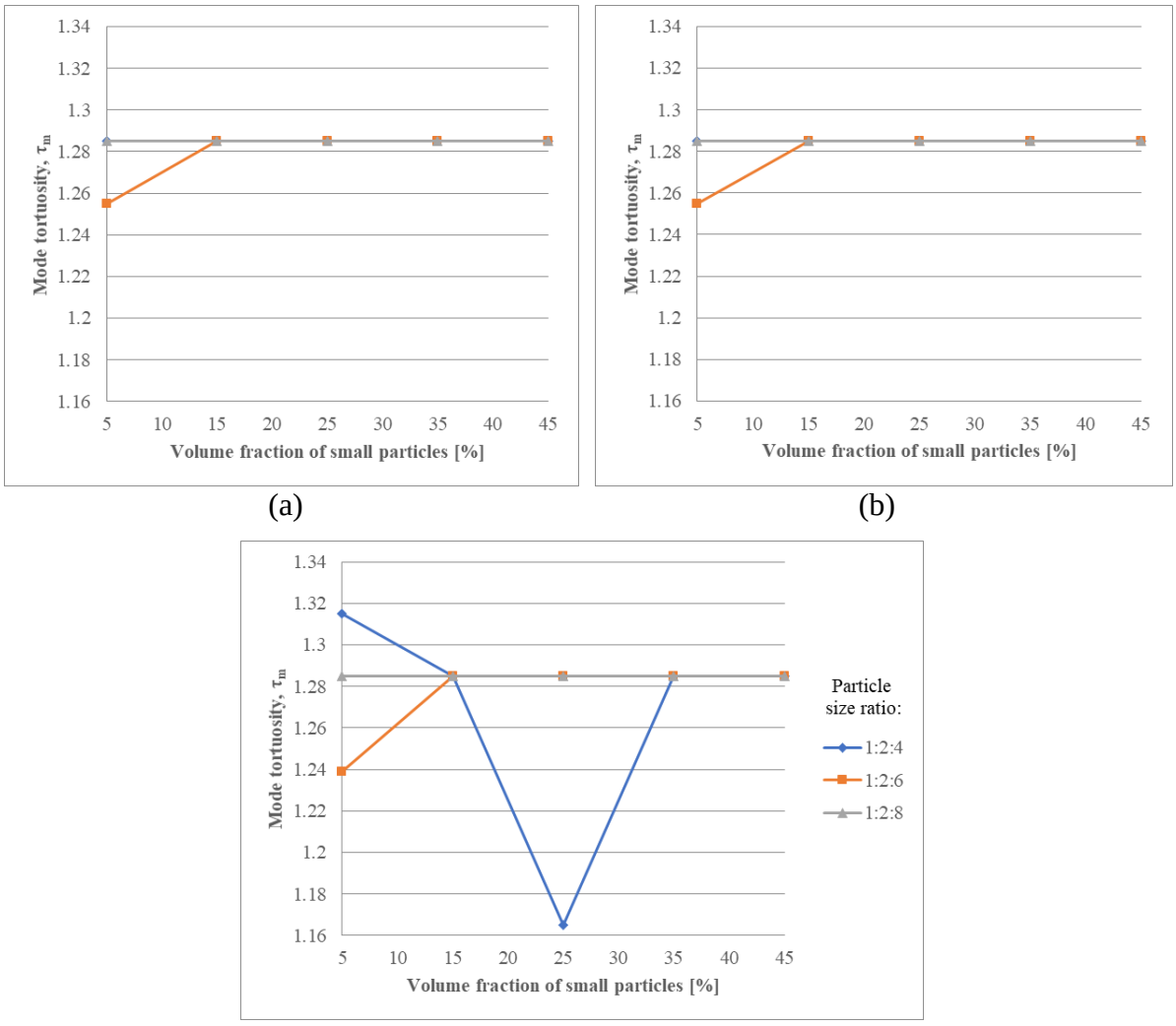
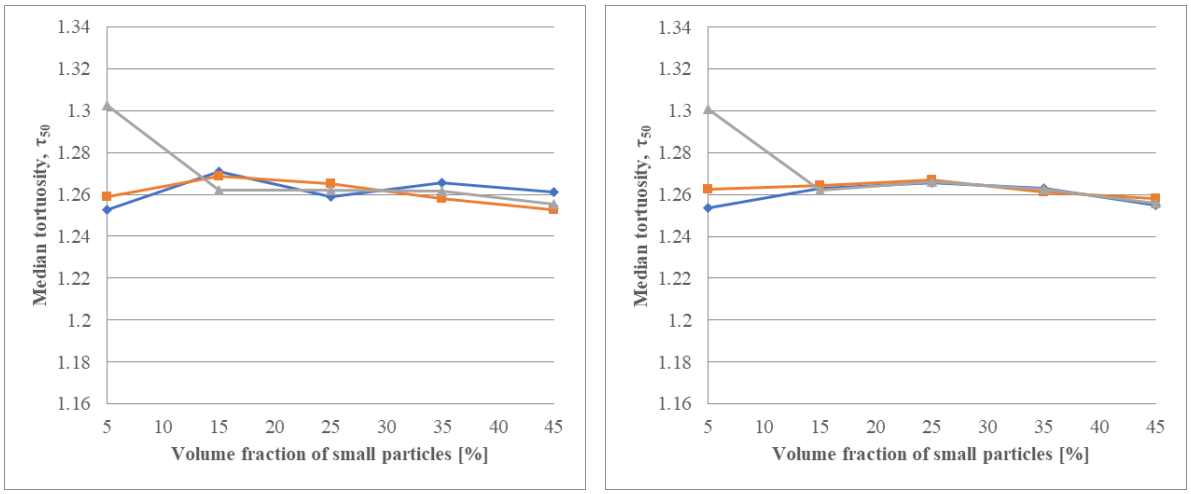
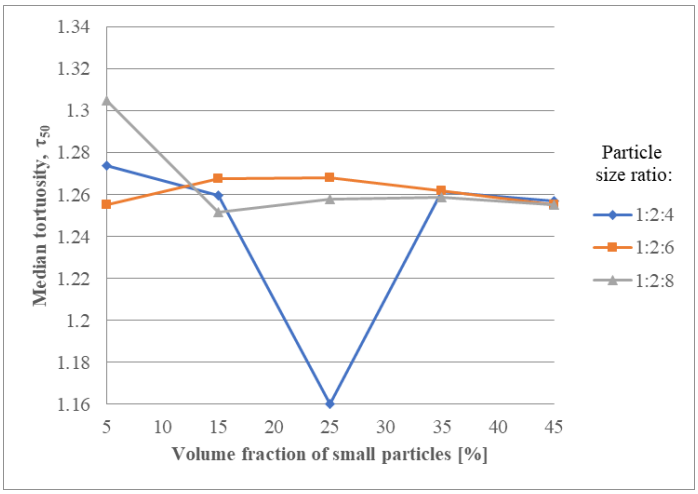


Figure 4.3: Mode tortuosity vs Volume fraction of small particles in: (a) x-, (b) y-, and (c) z-directions.



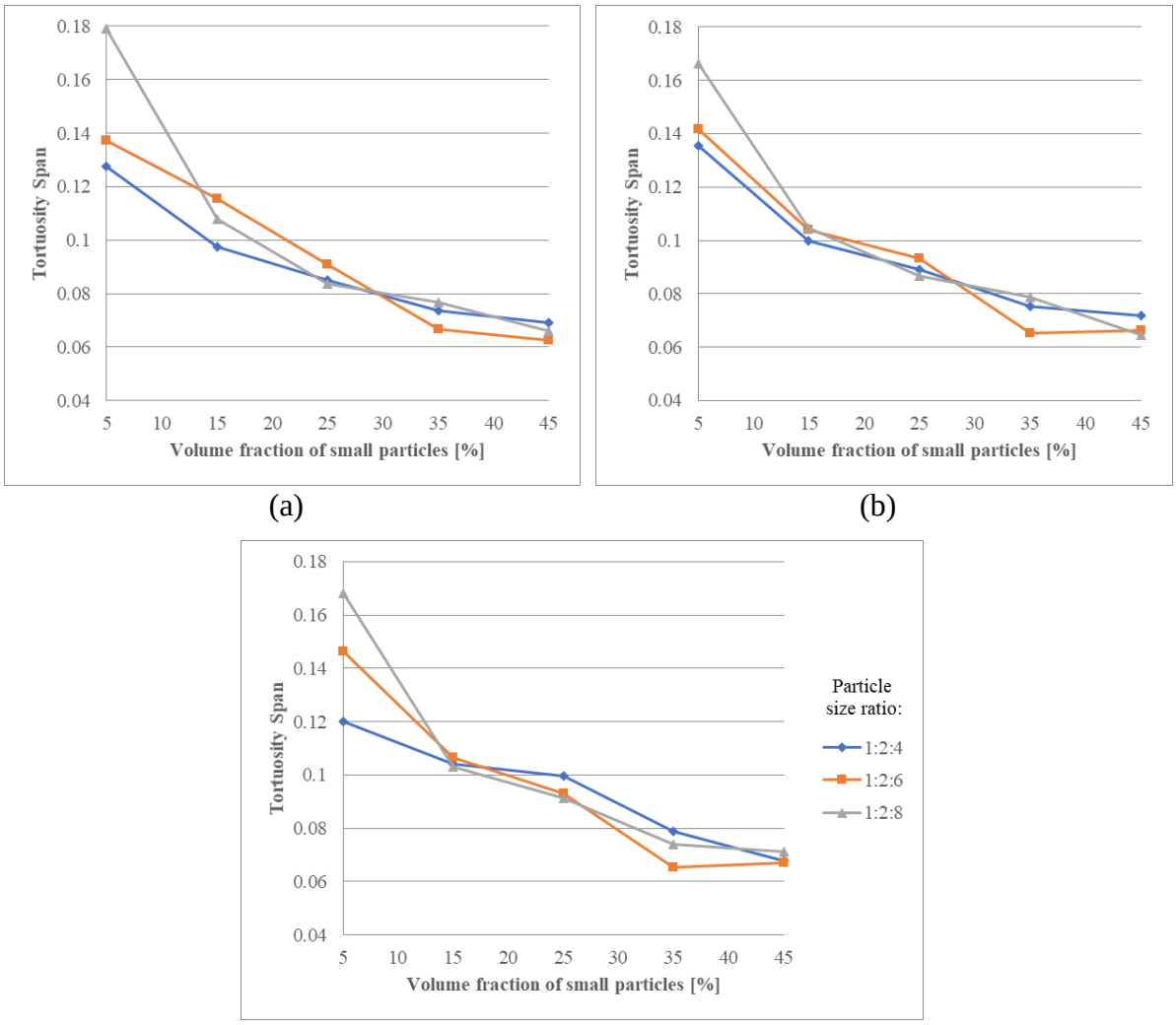
(a)

(b)



(c)

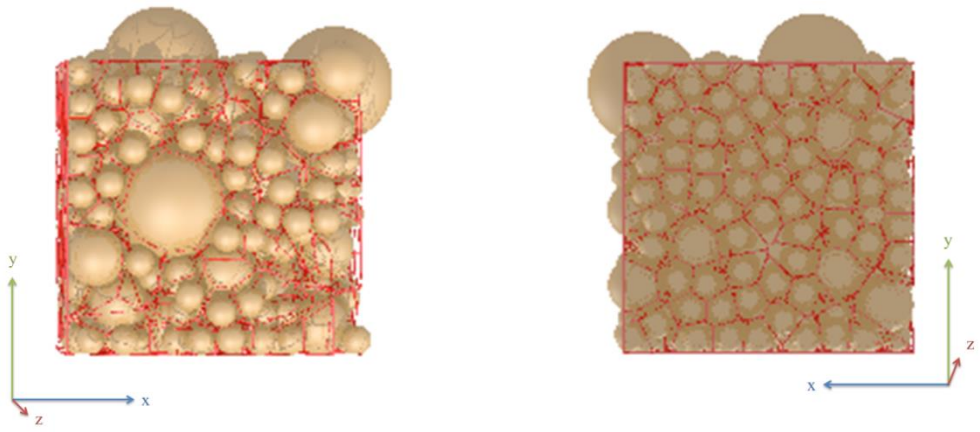
Figure 4.4: Median tortuosity vs Volume fraction of small particles in: (a) x-, (b) y-, and (c) z-directions.



(a) (b)

(c)

Figure 4.5: Tortuosity span vs Volume fraction of small particles in: (a) x-, (b) y-, and (c) z-directions.



(a) (b)

Figure 4.6: Shots of sample with particle size ratios 1:2:4 and fraction 25:25:50 from the (a) top and (b) bottom

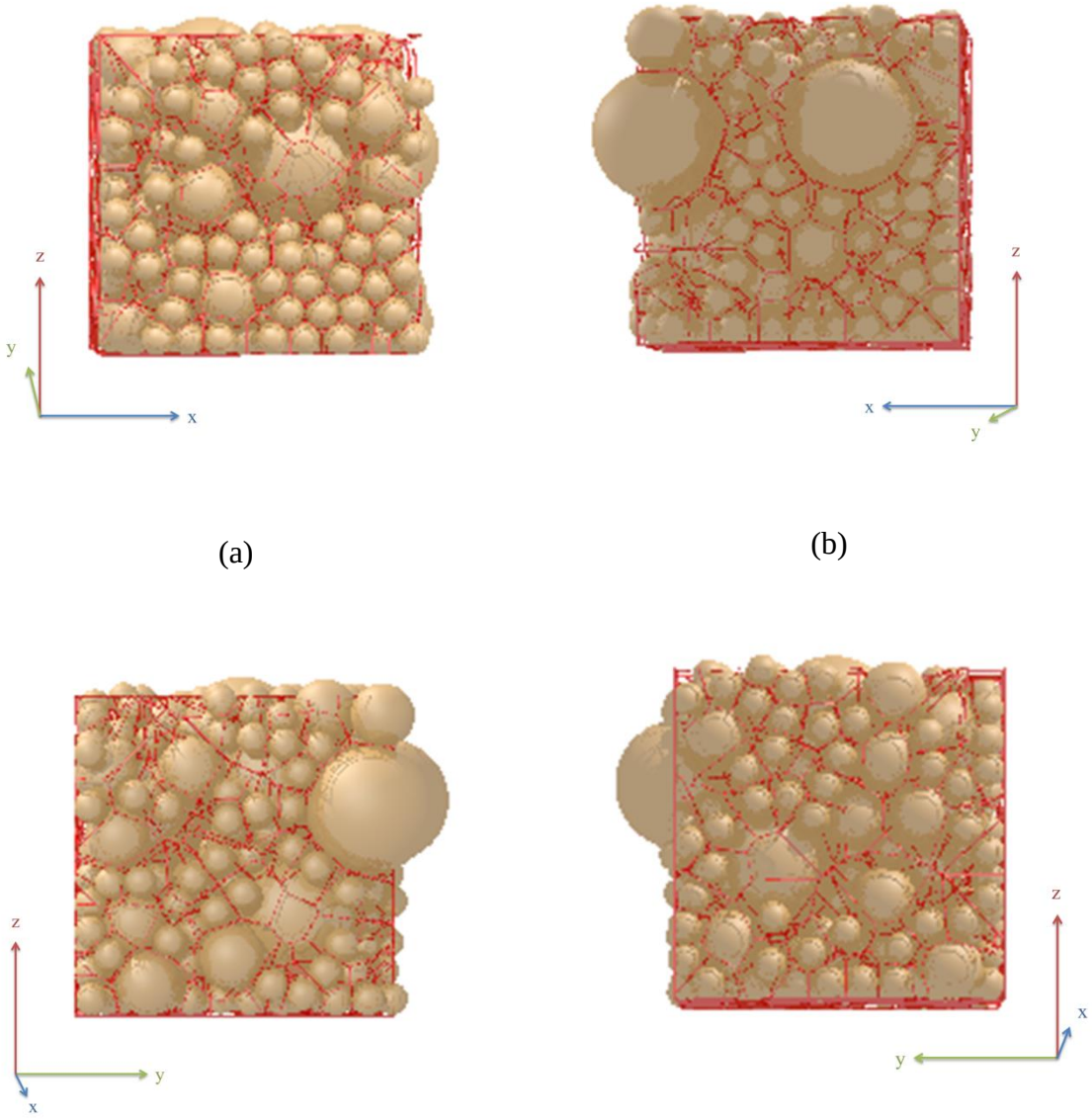


Figure 4.7: Shots of sample with fraction 25:25:50 and particle size ratios 1:2:4 from the sides

Likewise the previous case, for the case with particle size ratios 1:2:6 the distribution peak mainly lowers from roughly 36% to 18% by the addition of large particles, with some exceptions (Figure 4.8). Tortuosity value is distributed more widely for the samples with greater amount of large particles, 1.525 and 1.375 for samples with 90% and 10% large particles, respectively.

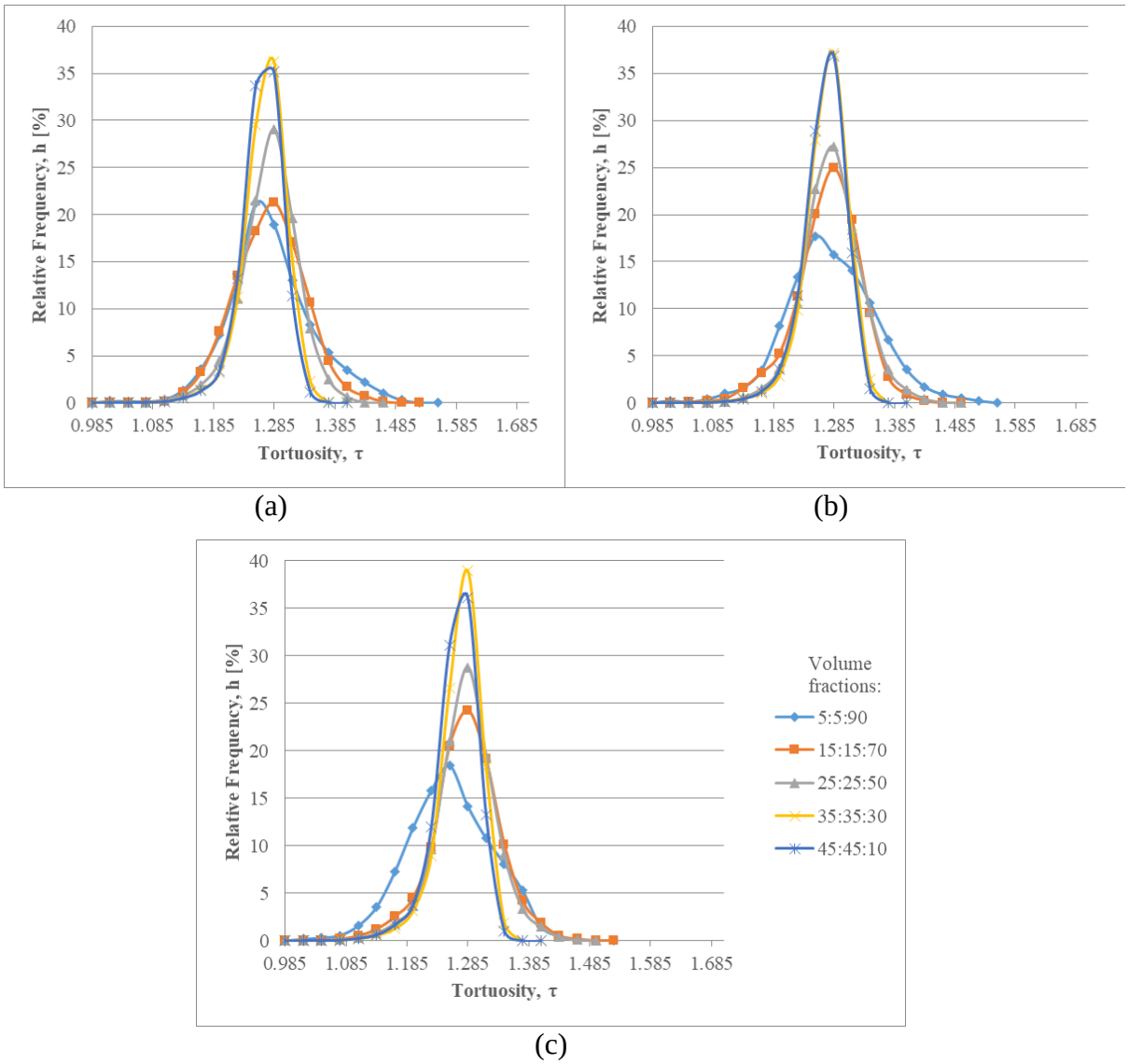
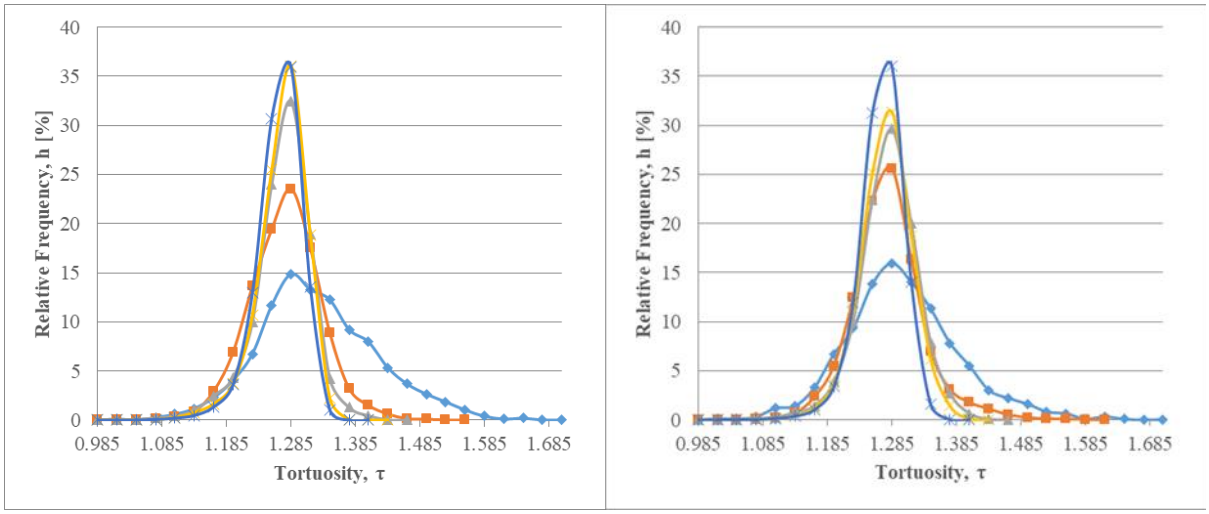
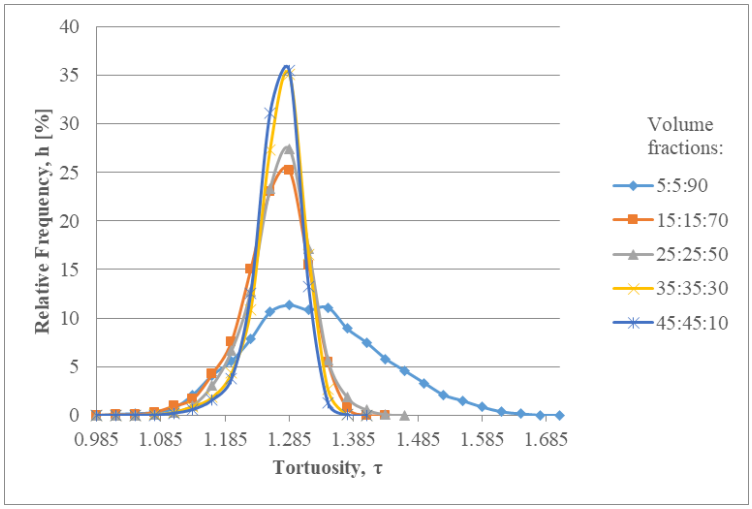


Figure 4.8: Relative frequency of tortuosity of ternary compacts with particle size ratios 1:2:6 and different volume fractions measured in: (a) x-direction, (b) y-direction, and (c) z-direction.

Concerning the mean tortuosity for the case with particle size ratio 1:2:6 in Figure 4.2, there is no considerable difference between the values, also being around 1.27 as in case with particle size ratio 1:2:4. Similar to the case with particle size ratio 1:2:4, the mode tortuosity (Figure 4.3) is equal to 1.285 for most samples, although the sample 5:5:90 have slightly different values of 1.239 in z- and 1.255 in x- and y-directions. Overall picture of the median tortuosity (Figure 4.4) for all samples of the case is also resembling the case 1:2:4 where the median value is around 1.26. The span (Figure 4.5) declines from nearly 0.14 to around 0.065 with the decrease of large particle volume fraction from 90% to 10%.



(a) (c)



(c)

Figure 4.9: Relative frequency of tortuosity of ternary compacts with particle size ratios 1:2:8 and different volume fractions measured in: (a) x-direction, (b) y-direction, and (c) z-direction.

As can be seen from Figure 4.9, peaks of tortuosity distribution for the case 1:2:8 samples decreases by the increase of large particle content, as in the previous cases, from 36% to 11%. This time the distribution of the sample 5:5:90 lasted even to tortuosity value of 1.705 which is much higher than previous cases, and remaining at 1.375 for sample 45:45:10. The tortuosity of the sample 5:5:90 has well defined right skewed distribution.

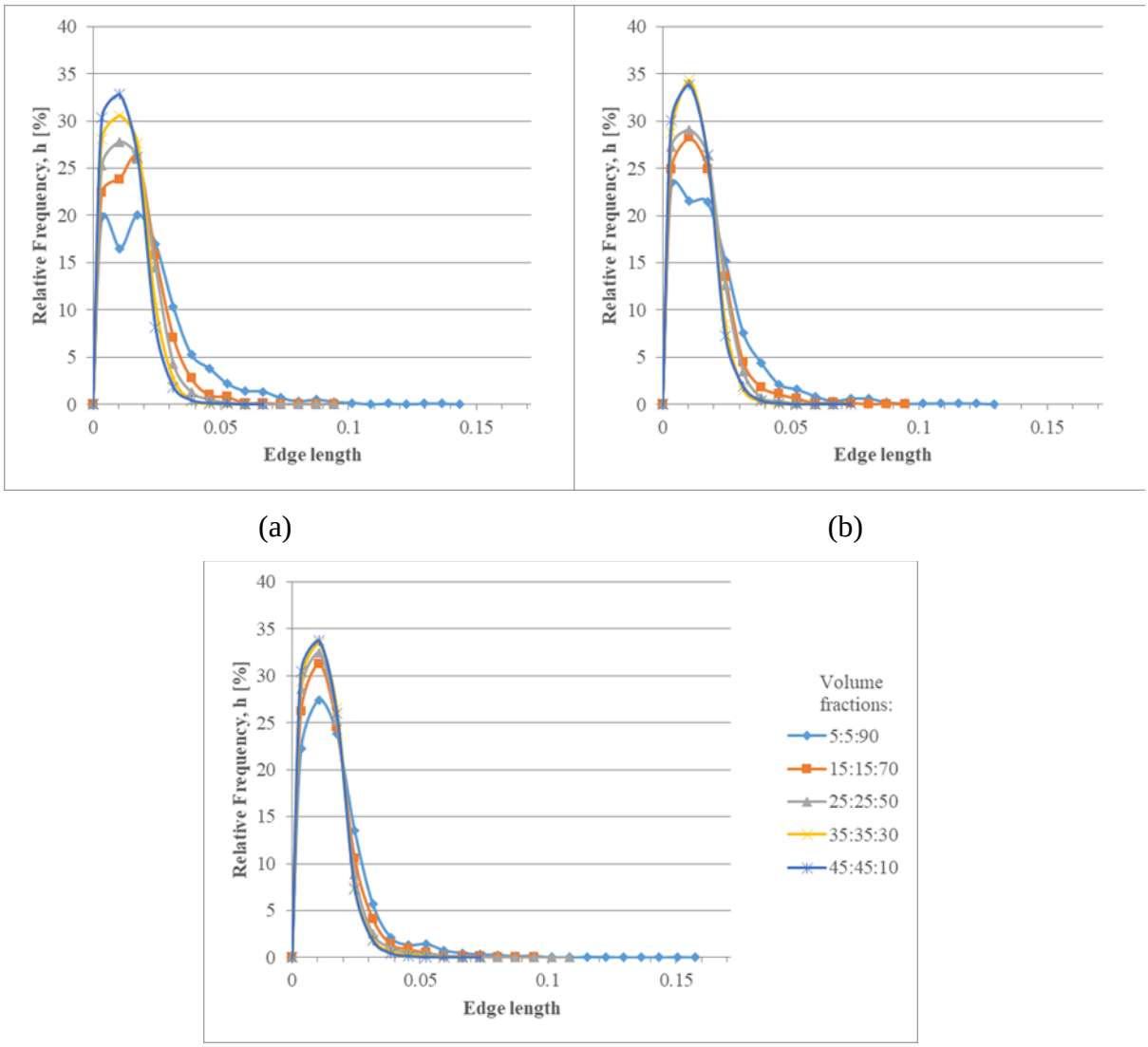
Talking about the parameters for the case 1:2:8, mean tortuosity (Figure 4.2) also fluctuates around 1.27, while it is worth noting that for the sample 5:5:90 the value is shifted to the right being nearly 1.32. According to the Figure 4.3, the mode tortuosity is

equal to 1.285 for all cases of the ternary mixtures. As for abovementioned cases the median tortuosity (Figure 4.4) is around 1.26, once again exception being the case 5-5-90 with the value of 1.3. Figure 4.5 shows that the span decreases from around 0.17 to 0.067 for when the volume fraction of large particles also goes down from 90% to 10%. It is worth to note that the sample 5:5:90 has mode of 1.285, median of around 1.3 and mean value 1.32, which is inherent arrangement for right skewness.

4.2 Relation between Voronoi parameters and ternary mixture parameters.

Figure 4.10 illustrates that the peak of edge length distribution also lowers by the growth of large particle amount in the mixture, as for tortuosity distribution. With increasing numbers of larger particles and their size, longer edges appear. Although the 5:5:90 sample of the case 1:2:8 has some longest edges, its distribution peak is sharper than in other cases.

For all cases of particle size ratio the peak of distribution of area per face goes down and the distribution range gets wider by the growth in number of large particle (Figure 4.11). With increasing size of a large particle, peaks of intermediate fractions gets sharper and increase from 41.239% to 60.086% for fraction 15:15:70 and from 77.532% to 83.583% for fraction 25:25:50. However distribution range also increases from 0.00247 to 0.00367 for fraction 5:5:90 and from 0.001125 to 0.002175 for fraction 15:15:70. Moreover, the distribution becomes more positively skewed for fraction 5:5:90 (the mode value shifts to the left from 0.000525 to 0.000225).



(a) (b)

(c)

Figure 4.10: Relative frequency of edge lengths of Voronoi polyhedrons for three cases of particle size ratio: (a) 1:2:4, (b) 1:2:6 and (c) 1:2:8 with different volume fractions.

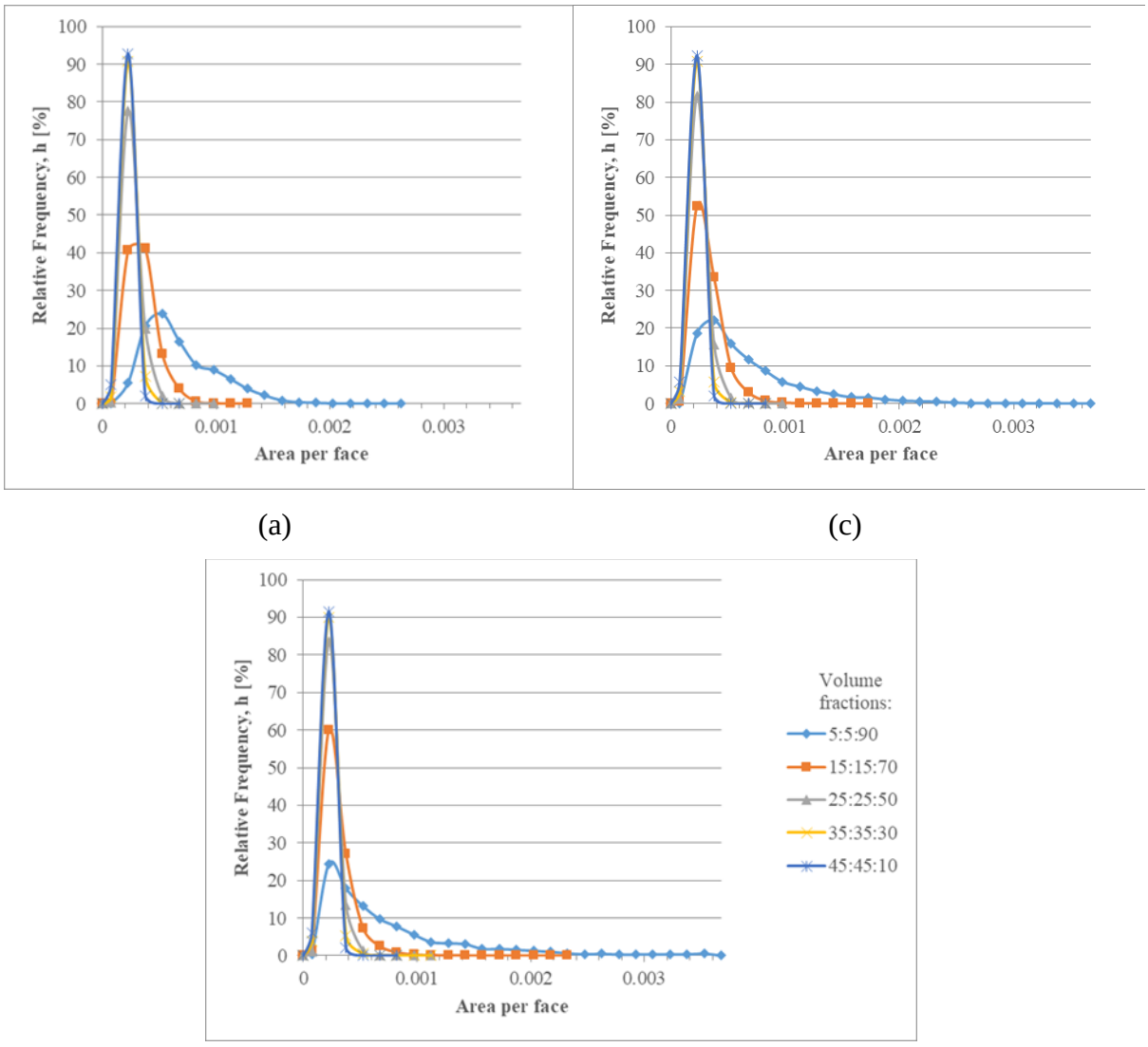


Figure 4.11: Relative frequency of area per face of Voronoi polyhedrons for three cases of particle size ratio: (a) 1:2:4, (b) 1:2:6 and (c) 1:2:8 with different volume fractions.

4.3 Summary

Overall, for all three cases of particle size ratio 1:2:4, 1:2:6 and 1:2:8 there is a trend that from fraction 5:5:90 to 45:45:10 the height of the tortuosity distribution peak rises. Thereby samples 5:5:90 have wider range of tortuosity values. This feature becomes more and more defined for particle size ratio cases 1:2:6 and 1:2:8 where the size of large particles becomes larger and larger, each time the tortuosity distribution becoming more asymmetric and positively skewed. However, for binary samples 10:0:90 and 90:0:10 it is in the opposite direction, what is more the heights of 10:0:90 peaks are around the heights of 45:45:10 and the heights of 90:0:10 are close to 5:5:90.

The aforementioned tortuosity trend associated with comparing different fractions having the same particle size ratios is also present for Voronoi cell edge lengths and face areas, where the parameters increase with increasing fraction of large particles. However, when comparing packings with different particle size ratios but with the same fractions, the face areas show the opposite tendency to have narrower, higher peaks in fractions 15:15:70 and 25:25:50, and a tendency of peaks in fraction 5:5:90 to shift to the left, when the size of a large particle increases. Meanwhile, this can be compensated by the appearance of higher values of the face areas and their increase in quantity.

Thus, the dependence of tortuosity on Voronoi parameters can be noted. With the appearance of longer edges and bigger face areas of Voronoi cell, tortuosity with large values increases, while an increase in shorter edges and smaller faces results in narrower tortuosity distribution with smaller and more uniform values. Thus, as from the equation (5) the diffusion coefficient is inversely proportional to the tortuosity, shorter edges and smaller faces, as well as higher volume fraction of smaller particles, results to a better diffusivity.

Chapter 5 – Conclusions and Recommendations

In conclusion, this thesis reflects the work including the literature review, methodology and results and discussion parts on analysis of tortuosity of porous ternary mixtures. The literature review is concentrated on available methods applied for evaluation of tortuosity of porous structures. Among them, the DEM was indicated as the one used for modeling the packing structure in this research. The method of applying the Voronoi diagram [14] was chosen for this work, considering its efficiency and lower time consumption comparing to the other methods.

The simulation part contains three main stages. First is the simulations of ternary powder compacts having different size ratios of spherical particles (r_{small}:r_{medium}:r_{large} 1:2:4, 1:2:6 and 1:2:8) and different volume fractions of particles (f_{small}:f_{medium}:f_{large} 5:5:90, 15:15:70, 25:25:50, 35:35:30 and 45:45:10) using DEM. Second, construction of radical Voronoi tessellation for the obtained packing structures in Voro++ software. Third, applying Dijkstra's algorithm to the Voronoi diagrams tortuosity distribution through modeled powder compact samples was calculated in all x, y and z-directions for each sample.

Study on relation between tortuosity and volume fractions of ternary mixture

In each of the three considered cases of particle size ratios (1:2:4, 1:2:6 and 1:2:8), the tortuosity distribution peak shows tendency to shorten and become less sharp during the increase in the volume fraction of large particles (from 10% to 90%). In this connection, higher tortuosity values appear in samples with 90% large particles, widening the range of the tortuosity distribution.

Study on relation between tortuosity and particle size ratio

As the size of large particles increases the feature of tortuosity distribution in samples with 90% large particles to become wider and have higher values becomes more distinct and having right skew.

Study on relation between tortuosity and coordinate direction

There is no a clear tendency that, with a change in the directions of the coordinates, the distribution of tortuosity values would change. However, for samples with fraction 5:5:90 of all particle ratios there are some deviations in tortuosity distributions by z-

direction, for each case of particle size ratio the tendency being different (or without having relation to alteration of particle size ratios). The tortuosity mode of the case 1:2:4 is shifted to the right for z-direction compared to x and y-directions from 1.285 to 1.315, in the contrary, for the case 1:2:6 it is left-shifted from 1.255 to 1.239, and for the case 1:2:8 it is stable at 1.285 for all directions. The deviations can be ascribed to the difference in the type of boundary conditions assigned on the planes orthogonal to x and y-directions (periodic) with z-direction (fixed). Additionally, there is only one sample (with particle size ratio 1:2:4 and fraction 25:25:50) with all tortuosity distribution parameters sharply shifted to the left in z-direction. The anomaly explained with an uneven arrangement due to gravity force while packing.

Study on relation between tortuosity and Voronoi parameters

Voronoi cell edge lengths and face areas showed the aforesaid tendency that tortuosity distribution peak goes up by the increase of small particle volume fraction. In the contrast, when comparing samples with different particle size ratios, by the increase of large particle in size, the face area distribution peak has the tendency to get higher and narrower in fractions 15:15:70 and 25:25:50, and shifts to the left in fraction 5:5:90. Meanwhile, this can be compensated by the appearance of higher values of the face areas and their increase in quantity.

Thus, the relation between tortuosity and Voronoi parameters can be described as follows. The distribution of tortuosity is wider and more positively skewed with the appearance of longer edges and bigger faces of Voronoi cells, and conversely smaller edge lengths and face areas lead to a narrower tortuosity distribution with smaller values. Following that diffusivity and tortuosity are inversely related, shorter edges and smaller faces are favorable in view reaching better diffusivity coefficient. Increasing the volume fraction of smaller particles assists better diffusivity, as well.

Future Recommendations

There is a need of further research works to be done. For example, validation of methodology by other non-geometric calculation methods such as flux based tortuosity calculations or experimentally, for example using FIB-SEM of porous structures, is necessary to evaluate the reliability of the applied method. Moreover, other more complicated shapes can also be analyzed to compare with the ternary mixtures. Additionally, other Voronoi parameters can also be considered for the analysis of their effect on the tortuosity. Further to this, in view of time efficiency, Dijkstra's algorithm,

used for finding the shortest path, can be compared with other algorithms to find a more efficient algorithm.

Bibliography

1. R. Xiong, Y. Duan, J. Cao and Q. Yu, "Battery and ultracapacitor in-the-loop approach to validate a real-time power management method for an all-climate electric vehicle", *Applied Energy*, vol. 217, pp. 153-165, 2018. doi: 10.1016/j.apenergy.2018.02.128.
2. T. Müllner, K. K. Unger, and U. Tallarek, "Characterization of microscopic disorder in reconstructed porous materials and assessment of mass transport-relevant structural descriptors", *New J. Chem.*, vol. 40, no. 5, pp. 3993–4015, 2016.
3. P. Braun, J. Cho, J. Pikul, W. King, and H. Zhang, "High power rechargeable batteries", *Current Opinion in Solid State and Materials Science*, vol. 16, no. 4, pp. 186-198, 2012. doi: 10.1016/j.cossms.2012.05.002.
4. D. Miranda, C. Costa, A. Almeida and S. Lanceros-Méndez, "Computer simulations of the influence of geometry in the performance of conventional and unconventional lithium-ion batteries", *Applied Energy*, vol. 165, pp. 318-328, 2016. doi: 10.1016/j.apenergy.2015.12.068.
5. S. Mitchell and M. Ortiz, "Computational multiobjective topology optimization of silicon anode structures for lithium-ion batteries", *Journal of Power Sources*, vol. 326, pp. 242-251, 2016. doi: 10.1016/j.jpowsour.2016.06.136.
6. V. Ramadesigan, P. Northrop, S. De, S. Santhanagopalan, R. Braatz and V. Subramanian, "Modeling and Simulation of Lithium-Ion Batteries from a Systems Engineering Perspective", *Journal of The Electrochemical Society*, vol. 159, no. 3, pp. R31-R45, 2012. doi: 10.1149/2.018203jes.
7. S. Lopez-Orozco, A. Inayat, A. Schwab, T. Selvam and W. Schwieger, "Zeolitic Materials with Hierarchical Porous Structures", *Advanced Materials*, vol. 23, no. 22-23, pp. 2602-2615, 2011. doi: 10.1002/adma.201100462.
8. Y. Li, Z. Fu and B. Su, "Hierarchically Structured Porous Materials for Energy Conversion and Storage", *Advanced Functional Materials*, vol. 22, no. 22, pp. 4634-4667, 2012. doi: 10.1002/adfm.201200591.
9. D. Hlushkou, A. Svidrytski and U. Tallarek, "Tracer-Size-Dependent Pore Space Accessibility and Long-Time Diffusion Coefficient in Amorphous, Mesoporous Silica", *The Journal of Physical Chemistry C*, vol. 121, no. 15, pp. 8416-8426, 2017. doi: 10.1021/acs.jpcc.7b00264.
10. M. Kishimto, H. Iwai, M. Saito and H. Yoshida, "Quantitative Evaluation of SOFC Electrode Micro structure by Diffusion Simulation Based on Random Walk Process", *The Proceedings of the Thermal Engineering Conference*, vol. 2009, no. 0, pp. 115-116, 2009. doi: 10.1299/jsmeted.2009.115.iw
11. H. Iwai et al., "Quantification of SOFC anode microstructure based on dual beam FIB-SEM technique", *Journal of Power Sources*, vol. 195, no. 4, pp. 955-961, 2010. doi: 10.1016/j.jpowsour.2009.09.005.
12. P. Jorgensen, K. Hansen, R. Larsen and J. Bowen, "Geometrical characterization of interconnected phase networks in three dimensions", *Journal of Microscopy*, vol. 244, no. 1, pp. 45-58, 2011. doi: 10.1111/j.1365-2818.2011.03504.x.
13. B. Tjaden, J. Lane, P. Withers, R. Bradley, D. Brett and P. Shearing, "The application of 3D imaging techniques, simulation and diffusion experiments to explore transport properties in porous oxygen transport membrane support materials", *Solid State Ionics*, vol. 288, pp. 315-321, 2016. doi: 10.1016/j.ssi.2016.01.030.

14. V. Semeykina, E. Malkovich, Y. Bazaikin, A. Lysikov and E. Parkhomchuk, "Optimal catalyst texture in macromolecule conversion: A computational and experimental study", *Chemical Engineering Science*, vol. 188, pp. 1-10, 2018. doi: 10.1016/j.ces.2018.05.005.
15. A. Einstein, "Investigations on the Theory of the Brownian Movement", *Annalen der Physik*, vol. 17, p. 549, 1905.
16. M. L. Lobanov, M. A. Zorina, *Methods for determining diffusion coefficients*, Yekaterinburg: Ural Publishing House, 2017.
17. H. Gao, Q. Wu, Y. Hu, J. Zheng, K. Amine and Z. Chen, "Revealing the Rate-Limiting Li-Ion Diffusion Pathway in Ultrathick Electrodes for Li-Ion Batteries", *The Journal of Physical Chemistry Letters*, vol. 9, no. 17, pp. 5100-5104, 2018. doi: 10.1021/acs.jpclett.8b02229.
18. Y. Zhu, H. Pham and J. Park, "A New Aspect of the Li Diffusion Enhancement Mechanism of Ultrathin Coating Layer on Electrode Materials", *ACS Applied Materials & Interfaces*, vol. 11, no. 42, pp. 38719-38726, 2019. doi: 10.1021/acsami.9b12740.
19. E. Maire and P. Withers, "Quantitative X-ray tomography", International Materials Reviews, vol. 59, no. 1, pp. 1-43, 2013. doi: 10.1179/1743280413y.0000000023.
20. M. Cantoni and L. Holzer, "Advances in 3D focused ion beam tomography", *MRS Bulletin*, vol. 39, no. 4, pp. 354-360, 2014. doi: 10.1557/mrs.2014.54.
21. H. Iwai et al., "Quantification of SOFC anode microstructure based on dual beam FIB-SEM technique", *Journal of Power Sources*, vol. 195, no. 4, pp. 955-961, 2010. doi: 10.1016/j.jpowsour.2009.09.005.
22. M. Kishimoto, H. Iwai, M. Saito, and H. Yoshida, "Quantitative Evaluation of Transport Properties of SOFC Porous Anode by Random Walk Process", *ECS Transactions*, vol. 24, no. 2, pp 1887-1896, 2009. doi: 10.1149/ma2009-02/12/1287.
23. C. Chueh, A. Bertei, J. Pharoah and C. Nicolella, "Effective conductivity in random porous media with convex and non-convex porosity", *International Journal of Heat and Mass Transfer*, vol. 71, pp. 183-188, 2014. doi: 10.1016/j.ijheatmasstransfer.2013.12.041.
24. D. Gostovic, J. Smith, D. Kundinger, K. Jones and E. Wachsman, "Three-Dimensional Reconstruction of Porous LSCF Cathodes", *Electrochemical and Solid-State Letters*, vol. 10, no. 12, p. B214, 2007. doi: 10.1149/1.2794672.
25. S. Cooper et al., "Image based modelling of microstructural heterogeneity in LiFePO 4 electrodes for Li-ion batteries", *Journal of Power Sources*, vol. 247, pp. 1033-1039, 2014. doi: 10.1016/j.jpowsour.2013.04.156.
26. D. Kehrwald, P. Shearing, N. Brandon, P. Sinha and S. Harris, "Local Tortuosity Inhomogeneities in a Lithium Battery Composite Electrode", *Journal of The Electrochemical Society*, vol. 158, no. 12, p. A1393, 2011. doi: 10.1149/2.079112jes.
27. D. Chung, M. Ebner, D. Ely, V. Wood and R. Edwin García, "Validity of the Bruggeman relation for porous electrodes", *Modelling and Simulation in Materials Science and Engineering*, vol. 21, no. 7, p. 074009, 2013. doi: 10.1088/0965-0393/21/7/074009.
28. P. Trogadas et al., "X-ray micro-tomography as a diagnostic tool for the electrode degradation in vanadium redox flow batteries", *Electrochemistry Communications*, vol. 48, pp. 155-159, 2014. doi: 10.1016/j.elecom.2014.09.010.
29. K. Soukup, P. Schneider and O. Šolcová, "Comparison of Wicke–Kallenbach and Graham's diffusion cells for obtaining transport characteristics of porous

solids", *Chemical Engineering Science*, vol. 63, no. 4, pp. 1003-1011, 2008. doi: 10.1016/j.ces.2007.10.032.
30. J. Yuan and B. Sundén, "On mechanisms and models of multi-component gas diffusion in porous structures of fuel cell electrodes", *International Journal of Heat and Mass Transfer*, vol. 69, pp. 358-374, 2014. doi: 10.1016/j.ijheatmasstransfer.2013.10.032.
31. S. Zhao, T. Evans, X. Zhou and S. Zhou, "Discrete element method investigation on thermally-induced shakedown of granular materials", *Granular Matter*, vol. 19, no. 1, p 11, 2016. doi: 10.1007/s10035-016-0690-5.
32. E. Dijkstra, "A note on two problems in connexion with graphs", Numerische Mathematik, vol. 1, no. 1, pp. 269-271, 1959. doi: 10.1007/bf01386390.
33. P. Cundall and O. Strack, "A discrete numerical model for granular assemblies", *Géotechnique*, vol. 29, no. 1, pp. 47-65, 1979. doi: 10.1680/geot.1979.29.1.47.
34. J. Ghaboussi and R. Barbosa, "Three-dimensional discrete element method for granular materials", *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 14, no. 7, pp. 451-472, 1990. doi: 10.1002/nag.1610140702.
35. R. Jensen, T. Edil, P. Bosscher, M. Plesha and N. Kahla, "Effect of Particle Shape on Interface Behavior of DEM-Simulated Granular Materials", *International Journal of Geomechanics*, vol. 1, no. 1, pp. 1-19, 2001. doi: 10.1061/(asce)1532-3641(2001)1:1(1).
36. R. Jensen, M. Plesha, T. Edil, P. Bosscher and N. Kahla, "DEM Simulation of Particle Damage in Granular Media — Structure Interfaces", *International Journal of Geomechanics*, vol. 1, no. 1, pp. 21-39, 2001. doi: 10.1061/(asce)1532-3641(2001)1:1(21).
37. LIGGGHTS®-PUBLIC, DCS Computing, Linz, Austria, retrieved from: https://www.cfdem.com/media/DEM/docu/Manual.html.
38. Z.Akhmetova, A. Boribayevaa, Z. Berkinovaa, A. Yermukhambetovaa, & B. Golmana, "Microstructural Features of Ternary Powder Compacts", *CHEMICAL ENGINEERING*, vol. 74, pp. 385-390, 2019. doi: 10.3303/CET1974065
39. C. Rycroft, "Voro++: a three-dimensional Voronoi cell library in C++", 2009. doi: 10.2172/946741.
40. S. Gass, M. Fu, "Encyclopedia of Operations Research and Management Science", *Encyclopedia of Operations Research and Management Science. Springer*. vol. 1., p. 428, 2013. doi: 10.1007/978-1-4419-1153-7.

Appendices

Appendix A. Input script for DEM simulation using LIGGGHTS

```
#Particle packing by insertion and successive growing of
particles

atom_style     granular
atom_modify    map array
boundary  p p f
newton         off
#echo          both

communicate    single vel yes

units          si

region         domain block 0.0 0.8 0.0 0.8 0. 3.0 units box
create_box     1 domain

neighbor  0.002 bin
neigh_modify   delay 0

timestep    0.0000005

fix gravi all gravity 9.81 vector 0.0 0.0 -1.0


#Material properties required for new pair styles

fix       m1 all property/global youngsModulus peratomtype
5.e7
fix       m2 all property/global poissonsRatio peratomtype
0.45
fix       m3 all property/global coefficientRestitution
peratomtypepair 1 0.3
fix       m4 all property/global coefficientFriction
peratomtypepair 1 0.5

#New pair style
pair_style gran model hertz tangential history  #Hertzian
without cohesion
pair_coeff     * *



fix xwalls1 all wall/gran model hertz tangential history
primitive type 1 xplane 0.00
```

```
fix xwalls2 all wall/gran model hertz tangential history
primitive type 1 xplane +0.8
fix ywalls1 all wall/gran model hertz tangential history
primitive type 1 yplane 0.00
fix ywalls2 all wall/gran model hertz tangential history
primitive type 1 yplane +0.8
fix zwalls1 all wall/gran model hertz tangential history
primitive type 1 zplane  0.00
fix zwalls2 all wall/gran model hertz tangential history
primitive type 1 zplane  3.0


#distributions for insertion
fix       pts1 all particletemplate/sphere 15485863 atom_type
1 density constant 2500 radius constant 0.06
fix             pts2 all particletemplate/sphere 15485867
atom_type 1 density constant 2500 radius constant 0.004
fix             pts3 all particletemplate/sphere 32452843
atom_type 1 density constant 2500 radius constant 0.003
fix             pts4 all particletemplate/sphere 49979687
atom_type 1 density constant 2500 radius constant 0.02
fix             pts5 all particletemplate/sphere 49979693
atom_type 1 density constant 2500 radius constant 0.01

fix       pdd1 all particledistribution/discrete 67867967 2
pts1 0.1 pts5 0.9

#particle insertion

#fix    ins all insert/stream seed 32452867
distributiontemplate pdd1 &
       nparticles 6000 massrate 0.1 insert_every 1000
overlapcheck yes all_in no vel constant 0.0 0.0 -1.0 &
       insertion_face inface

fix       ins all insert/pack  seed 67867979
distributiontemplate pdd1 &
               maxattempt 200 insert_every once overlapcheck
yes all_in yes vel constant 0. 0. 0. region domain
volumefraction_region 0.3

#apply nve integration to all particles that are inserted as
single particles
fix       integr all nve/sphere

# apply vibration
#fix move all move/mesh mesh cad wiggle amplitude 0.013 0.0
0.0 period 0.2
#fix walls all wall/gran model hertz tangential history mesh
#yes n meshes 1 meshes cad
```

```
#fix walls all wall/gran model hertz/stiffness tangential
history mesh store force

#region and insertion
group          all region domain

#output settings, include total thermal energy
#compute       1 all erotate/sphere
#thermo_style  custom step atoms ke c_1 vol
thermo         1000
#thermo_modify lost ignore norm no

#insert the first particles
run       1
dump      dmp all custom/vtk 3000 post/packing_*.vtk id type
type x y z ix iy iz vx vy vz fx fy fz omegax omegay omegaz
radius
dump            4a all custom 10000 post/packing_*. x y z
radius

#run simulation
restart 100000 part.restart
run         7000000 upto
```

Appendix B. Script for Voronoi tessellation using Voro++

```
#include "voro++.hh"
using namespace voro;
const double x_min1=0,x_max1=0.2;
const double y_min1=0,y_max1=0.2;
const double z_min1=0,z_max1=0.2;


const int n_x=6,n_y=6,n_z=6;

int main() {
     //container
con(x_min,x_max,y_min,y_max,z_min,z_max,n_x,n_y,n_z,
     //        false,false,false,1);

     container_poly
con(x_min1,x_max1,y_min1,y_max1,z_min1,z_max1,n_x,n_y,n_z,
               false,false,false,1);

     con.import("packing");
     con.draw_cells_gnuplot("p1_v.gnu");
     con.draw_particles("p1_p.gnu");
     con.draw_particles_pov("p1_p.pov");
     con.draw_cells_pov("p1_v.pov");
```

```
     con.draw_particles_pov("packing_periodic_1.pov");

     con.print_custom("%i %x %y %z %r %i %w %g %s %E %F
%v","p1_main.output");
     con.print_custom("%i %m %p %o","p1_vertex.output");
     con.print_custom("%i %w %P %s %f","p1_face.output");
     con.print_custom("%i %s %a %E
%g","p1_face_edge.output");
     con.print_custom("%i %s %e","p1_face_perimeter.output");
     con.print_custom("%i %q %r %s %F %A","wj_1.output");
     con.print_custom("%i %q %r %s %F %a","wj_2.output");
     con.print_custom("%i %q %r %s %F %f","wj_3.output");
     con.print_custom("%i %q %r %s %F %t","wj_4.output");
     con.print_custom("%i %q %r %s %F %l","wj_5.output");
     con.print_custom("%i %q %r %s %F %n","wj_6.output");}
```

Appendix C. Matlab simulation file for tortuosity calculation

```
close all
clear
clc
nodes=[1,0.0,0.0,0.0
2,0.0,0.0,0.0230997
3,0.0287886,0.0148436,0.0165645
…];
segments=[1,2,1
2,4,1
3,5,2
…];
s=[];
f=[];
p=size(nodes, 1);
for t=1:p
   if nodes(t,2)==0
      s=[s; nodes(t,:)];
   elseif nodes(t,2)==0.2
      f=[f; nodes(t,:)];
   end
end
```

```
n=size(s, 1);
k=size(f, 1);
distan=[];
eucl_distan=[];
tortuos=[];
for i=1:n
   start_id=s(i,1);
   for j=1:k
      finish_id=f(j,1);
      [distance,path] = dijkstra(nodes,segments,start_id,finish_id);
      distan=[distan; distance];
      eucl_dist=sqrt((f(j,2)-s(i,2))^2+(f(j,3)-s(i,3))^2+(f(j,4)-s(i,4))^2);
      eucl_distan=[eucl_distan; eucl_dist];
      tortuosity=distance/eucl_dist;
      tortuos=[tortuos; tortuosity];
   end
end
disp('distance=');
disp(num2str(distan));
disp('euc_d=');
disp(num2str(eucl_distan));
disp('t=');
disp(num2str(tortuos));
mean_t=sum(tortuos)/(n*k);
disp(['mean_t=' num2str(mean_t)]);

m=size(tortuos,1);
bin1=[];
for i=1:m
   if tortuos(i,1)<=1.03
      bin1=[bin1; tortuos(i,1)];
   end
end
bin2=[];
```

```
for i=1:m
   if tortuos(i,1)>1.03 & tortuos(i,1)<=1.06
      bin2=[bin2; tortuos(i,1)];
   end
end
bin3=[];
for i=1:m
   if tortuos(i,1)>1.06 & tortuos(i,1)<=1.09
      bin3=[bin3; tortuos(i,1)];
   end
end
bin4=[];
for i=1:m
   if tortuos(i,1)>1.09 & tortuos(i,1)<=1.12
      bin4=[bin4; tortuos(i,1)];
   end
end
bin5=[];
for i=1:m
   if tortuos(i,1)>1.12 & tortuos(i,1)<=1.15
      bin5=[bin5; tortuos(i,1)];
   end
end
bin6=[];
for i=1:m
   if tortuos(i,1)>1.15 & tortuos(i,1)<=1.18
      bin6=[bin6; tortuos(i,1)];
   end
end
bin7=[];
for i=1:m
   if tortuos(i,1)>1.18 & tortuos(i,1)<=1.21
      bin7=[bin7; tortuos(i,1)];
   end
```

```
end
bin8=[];
for i=1:m
   if tortuos(i,1)>1.21 & tortuos(i,1)<=1.24
      bin8=[bin8; tortuos(i,1)];
   end
end
bin9=[];
for i=1:m
   if tortuos(i,1)>1.24 & tortuos(i,1)<=1.27
      bin9=[bin9; tortuos(i,1)];
   end
end
bin10=[];
for i=1:m
   if tortuos(i,1)>1.27 & tortuos(i,1)<=1.30
      bin10=[bin10; tortuos(i,1)];
   end
end
bin11=[];
for i=1:m
   if tortuos(i,1)>1.30 & tortuos(i,1)<=1.33
      bin11=[bin11; tortuos(i,1)];
   end
end
bin12=[];
for i=1:m
   if tortuos(i,1)>1.33 & tortuos(i,1)<=1.36
      bin12=[bin12; tortuos(i,1)];
   end
end
bin13=[];
for i=1:m
   if tortuos(i,1)>1.36 & tortuos(i,1)<=1.39
```

```
        bin13=[bin13; tortuos(i,1)];
    end
end
bin14=[];
for i=1:m
    if tortuos(i,1)>1.39 & tortuos(i,1)<=1.42
        bin14=[bin14; tortuos(i,1)];
    end
end
bin15=[];
for i=1:m
    if tortuos(i,1)>1.42 & tortuos(i,1)<=1.45
        bin15=[bin15; tortuos(i,1)];
    end
end
bin16=[];
for i=1:m
    if tortuos(i,1)>1.45 & tortuos(i,1)<=1.48
        bin16=[bin16; tortuos(i,1)];
    end
end
bin17=[];
for i=1:m
    if tortuos(i,1)>1.48 & tortuos(i,1)<=1.51
        bin17=[bin17; tortuos(i,1)];
    end
end
bin18=[];
for i=1:m
    if tortuos(i,1)>1.51 & tortuos(i,1)<=1.54
        bin18=[bin18; tortuos(i,1)];
    end
end
bin19=[];
```

```
for i=1:m
   if tortuos(i,1)>1.54 & tortuos(i,1)<=1.57
      bin19=[bin19; tortuos(i,1)];
   end
end
bin20=[];
for i=1:m
   if tortuos(i,1)>1.57 & tortuos(i,1)<=1.6
      bin20=[bin20; tortuos(i,1)];
   end
end
bin21=[];
for i=1:m
   if tortuos(i,1)>1.6 & tortuos(i,1)<=1.63
      bin21=[bin21; tortuos(i,1)];
   end
end
bin22=[];
for i=1:m
   if tortuos(i,1)>1.63 & tortuos(i,1)<=1.66
      bin22=[bin22; tortuos(i,1)];
   end
end
bin23=[];
for i=1:m
   if tortuos(i,1)>1.66 & tortuos(i,1)<=1.69
      bin23=[bin23; tortuos(i,1)];
   end
end
bin24=[];
for i=1:m
   if tortuos(i,1)>1.69 & tortuos(i,1)<=1.72
      bin24=[bin24; tortuos(i,1)];
   end
```

```
end
disp(['bsize1=', num2str(size(bin1,1))]);
disp(['bsize2=', num2str(size(bin2,1))]);
disp(['bsize3=', num2str(size(bin3,1))]);
disp(['bsize4=', num2str(size(bin4,1))]);
disp(['bsize5=', num2str(size(bin5,1))]);
disp(['bsize6=', num2str(size(bin6,1))]);
disp(['bsize7=', num2str(size(bin7,1))]);
disp(['bsize8=', num2str(size(bin8,1))]);
disp(['bsize9=', num2str(size(bin9,1))]);
disp(['bsize10=', num2str(size(bin10,1))]);
disp(['bsize11=', num2str(size(bin11,1))]);
disp(['bsize12=', num2str(size(bin12,1))]);
disp(['bsize13=', num2str(size(bin13,1))]);
disp(['bsize14=', num2str(size(bin14,1))]);
disp(['bsize15=', num2str(size(bin15,1))]);
disp(['bsize16=', num2str(size(bin16,1))]);
disp(['bsize17=', num2str(size(bin17,1))]);
disp(['bsize18=', num2str(size(bin18,1))]);
disp(['bsize19=', num2str(size(bin19,1))]);
disp(['bsize20=', num2str(size(bin20,1))]);
disp(['bsize21=', num2str(size(bin21,1))]);
disp(['bsize22=', num2str(size(bin22,1))]);
disp(['bsize23=', num2str(size(bin23,1))]);
disp(['bsize24=', num2str(size(bin24,1))]);
disp(['m=', num2str(m)]);
disp(['max=', num2str(max(tortuos))]);
disp(num2str(size(bin1,1)));
disp(num2str(size(bin2,1)));
disp(num2str(size(bin3,1)));
disp(num2str(size(bin4,1)));
disp(num2str(size(bin5,1)));
disp(num2str(size(bin6,1)));
disp(num2str(size(bin7,1)));
```

```
disp(num2str(size(bin8,1)));
disp(num2str(size(bin9,1)));
disp(num2str(size(bin10,1)));
disp(num2str(size(bin11,1)));
disp(num2str(size(bin12,1)));
disp(num2str(size(bin13,1)));
disp(num2str(size(bin14,1)));
disp(num2str(size(bin15,1)));
disp(num2str(size(bin16,1)));
disp(num2str(size(bin17,1)));
disp(num2str(size(bin18,1)));
disp(num2str(size(bin19,1)));
disp(num2str(size(bin20,1)));
disp(num2str(size(bin21,1)));
disp(num2str(size(bin22,1)));
disp(num2str(size(bin23,1)));
disp(num2str(size(bin24,1)));
disp(num2str(m));
```

Appendix D. Matlab file for shortest pass calculation

```
function [dist,path] = dijkstra(nodes,segments,start_id,finish_id)
%DIJKSTRA Calculates the shortest distance and path between points on a map
%   using Dijkstra's Shortest Path Algorithm
%
% [DIST, PATH] = DIJKSTRA(NODES, SEGMENTS, SID, FID)
%    Calculates the shortest distance and path between start and finish nodes SID and FID
%
% [DIST, PATH] = DIJKSTRA(NODES, SEGMENTS, SID)
%   Calculates the shortest distances and paths from the starting node SID to all
%    other nodes in the map
%
```

```
% Note:
%    DIJKSTRA is set up so that an example is created if no inputs are provided,
%     but ignores the example and just processes the inputs if they are given.
%
% Inputs:
%    NODES should be an Nx3 or Nx4 matrix with the format [ID X Y] or [ID X Y
Z]
%     where ID is an integer, and X, Y, Z are cartesian position coordinates)
%    SEGMENTS should be an Mx3 matrix with the format [ID N1 N2]
%     where ID is an integer, and N1, N2 correspond to node IDs from NODES list
%      such that there is an [undirected] edge/segment between node N1 and node
N2
%     SID should be an integer in the node ID list corresponding with the starting
node
%     FID (optional) should be an integer in the node ID list corresponding with the
finish
%
% Outputs:
%    DIST is the shortest Euclidean distance
%     If FID was specified, DIST will be a 1x1 double representing the shortest
%       Euclidean distance between SID and FID along the map segments. DIST
will have
%      a value of INF if there are no segments connecting SID and FID.
%     If FID was not specified, DIST will be a 1xN vector representing the shortest
%      Euclidean distance between SID and all other nodes on the map. DIST will
have
%       a value of INF for any nodes that cannot be reached along segments of the
map.
%    PATH is a list of nodes containing the shortest route
%      If FID was specified, PATH will be a 1xP vector of node IDs from SID to
FID.
%      NAN will be returned if there are no segments connecting SID to FID.
%     If FID was not specified, PATH will be a 1xN cell of vectors representing the
```

```
%          shortest route from SID to all other nodes on the map. PATH will have a value
%          of NAN for any nodes that cannot be reached along the segments of the map.
%
% Example:
%    dijkstra; % calculates shortest path and distance between two nodes
%              % on a map of randomly generated nodes and segments
%
% Example:
%    nodes = [(1:10); 100*rand(2,10)]';
%    segments = [(1:17); floor(1:0.5:9); ceil(2:0.5:10)]';
%    figure; plot(nodes(:,2), nodes(:,3),'k.');
%    hold on;
%    for s = 1:17
%       if (s <= 10) text(nodes(s,2),nodes(s,3),[' ' num2str(s)]); end
%       plot(nodes(segments(s,2:3)',2),nodes(segments(s,2:3)',3),'k');
%    end
%    [d, p] = dijkstra(nodes, segments, 1, 10)
%    for n = 2:length(p)
%       plot(nodes(p(n-1:n),2),nodes(p(n-1:n),3),'r-.','linewidth',2);
%    end
%    hold off;
%
% Author: Joseph Kirk
% Email: jdkirk630 at gmail dot com
% Release: 1.3
% Release Date: 5/18/07

if (nargin < 3) % SETUP
   % (GENERATE RANDOM EXAMPLE OF NODES AND SEGMENTS IF NOT GIVEN AS INPUTS)
   % Create a random set of nodes/vertices,and connect some of them with
   % edges/segments. Then graph the resulting map.
```

```
num_nodes = 40; L = 100; max_seg_length = 30; ids = (1:num_nodes)';
nodes = [ids L*rand(num_nodes,2)]; % create random nodes
h = figure; plot(nodes(:,2),nodes(:,3),'k.') % plot the nodes
text(nodes(num_nodes,2),nodes(num_nodes,3),...
   [' ' num2str(ids(num_nodes))],'Color','b','FontWeight','b')
hold on
num_segs = 0; segments = zeros(num_nodes*(num_nodes-1)/2,3);
for i = 1:num_nodes-1 % create edges between some of the nodes
   text(nodes(i,2),nodes(i,3),[' ' num2str(ids(i))],'Color','b','FontWeight','b')
   for j = i+1:num_nodes
      d = sqrt(sum((nodes(i,2:3) - nodes(j,2:3)).^2));
      if and(d < max_seg_length,rand < 0.6)
         plot([nodes(i,2) nodes(j,2)],[nodes(i,3) nodes(j,3)],'k.-')
         % add this link to the segments list
         num_segs = num_segs + 1;
         segments(num_segs,:) = [num_segs nodes(i,1) nodes(j,1)];
      end
   end
end
segments(num_segs+1:num_nodes*(num_nodes-1)/2,:) = [];
axis([0 L 0 L])
% Calculate Shortest Path Using Dijkstra's Algorithm
% Get random starting/ending nodes,compute the shortest distance and path.
start_id = ceil(num_nodes*rand); disp(['start id = ' num2str(start_id)]);
finish_id = ceil(num_nodes*rand); disp(['finish id = ' num2str(finish_id)]);
[distance,path] = dijkstra(nodes,segments,start_id,finish_id);
disp(['distance = ' num2str(distance)]); disp(['path = [' num2str(path) ']']);
% If a Shortest Path exists,Plot it on the Map.
figure(h)
for k = 2:length(path)
   m = find(nodes(:,1) == path(k-1));
   n = find(nodes(:,1) == path(k));
   plot([nodes(m,2) nodes(n,2)],[nodes(m,3) nodes(n,3)],'ro-','LineWidth',2);
end
```

```
    title(['Shortest Distance from ' num2str(start_id) ' to ' ...
        num2str(finish_id) ' = ' num2str(distance)])
    hold off

else %---------------------------------------------------------------------------
    % MAIN FUNCTION - DIJKSTRA'S ALGORITHM

    % initializations
    node_ids = nodes(:,1);
    [num_map_pts,cols] = size(nodes);
    table = sparse(num_map_pts,2);
    shortest_distance = Inf(num_map_pts,1);
    settled = zeros(num_map_pts,1);
    path = num2cell(NaN(num_map_pts,1));
    col = 2;
    pidx = find(start_id == node_ids);
    shortest_distance(pidx) = 0;
    table(pidx,col) = 0;
    settled(pidx) = 1;
    path(pidx) = {start_id};
    if (nargin < 4) % compute shortest path for all nodes
        while_cmd = 'sum(~settled) > 0';
    else % terminate algorithm early
        while_cmd = 'settled(zz) == 0';
        zz = find(finish_id == node_ids);
    end
    while eval(while_cmd)
        % update the table
        table(:,col-1) = table(:,col);
        table(pidx,col) = 0;
        % find neighboring nodes in the segments list
        neighbor_ids = [segments(node_ids(pidx) == segments(:,2),3);
            segments(node_ids(pidx) == segments(:,3),2)];
        % calculate the distances to the neighboring nodes and keep track of the paths
```

```
        for k = 1:length(neighbor_ids)
            cidx = find(neighbor_ids(k) == node_ids);
            if ~settled(cidx)
                d = sqrt(sum((nodes(pidx,2:cols) - nodes(cidx,2:cols)).^2));
                if (table(cidx,col-1) == 0) || ...
                        (table(cidx,col-1) > (table(pidx,col-1) + d))
                    table(cidx,col) = table(pidx,col-1) + d;
                    tmp_path = path(pidx);
                    path(cidx) = {[tmp_path{1} neighbor_ids(k)]};
                else
                    table(cidx,col) = table(cidx,col-1);
                end
            end
        end
        % find the minimum non-zero value in the table and save it
        nidx = find(table(:,col));
        ndx = find(table(nidx,col) == min(table(nidx,col)));
        if isempty(ndx)
            break
        else
            pidx = nidx(ndx(1));
            shortest_distance(pidx) = table(pidx,col);
            settled(pidx) = 1;
        end
    end
    if (nargin < 4) % return the distance and path arrays for all of the nodes
        dist = shortest_distance';
        path = path';
    else % return the distance and path for the ending node
        dist = shortest_distance(zz);
        path = path(zz);
        path = path{1};
    end
end
```

Appendix E. Matlab file for edge length calculation

```
close all
clear
clc
nodes=[1,0.0,0.0,0.0
2,0.0,0.0,0.0230997
3,0.0287886,0.0148436,0.0165645
…];
segments=[1,2,1
2,4,1
3,5,2
…];
n=size(segments, 1);
lengthes=[];
for i=1:n
   s_id=segments(i,2);
   f_id=segments(i,3);
   dist=sqrt((nodes(f_id,2)-nodes(s_id,2))^2+(nodes(f_id,3)-
nodes(s_id,3))^2+(nodes(f_id,4)-nodes(s_id,4))^2);
   lengthes=[lengthes; dist];
end
m=size(lengthes,1);
bin1=[];
for j=1:m
   if lengthes(j,1)<=0.007
      bin1=[bin1; lengthes(j,1)];
   end
end
bin2=[];
for j=1:m
   if lengthes(j,1)>0.007 & lengthes(j,1)<=0.014
```

```
        bin2=[bin2; lengthes(j,1)];
    end
end
bin3=[];
for j=1:m
    if lengthes(j,1)>0.014 & lengthes(j,1)<=0.021
        bin3=[bin3; lengthes(j,1)];
    end
end
bin4=[];
for j=1:m
    if lengthes(j,1)>0.021 & lengthes(j,1)<=0.028
        bin4=[bin4; lengthes(j,1)];
    end
end
bin5=[];
for j=1:m
    if lengthes(j,1)>0.028 & lengthes(j,1)<=0.035
        bin5=[bin5; lengthes(j,1)];
    end
end
bin6=[];
for j=1:m
    if lengthes(j,1)>0.035 & lengthes(j,1)<=0.042
        bin6=[bin6; lengthes(j,1)];
    end
end
bin7=[];
for j=1:m
    if lengthes(j,1)>0.042 & lengthes(j,1)<=0.049
        bin7=[bin7; lengthes(j,1)];
    end
end
bin8=[];
```

```
for j=1:m
   if lengthes(j,1)>0.049 & lengthes(j,1)<=0.056
      bin8=[bin8; lengthes(j,1)];
   end
end
bin9=[];
for j=1:m
   if lengthes(j,1)>0.056 & lengthes(j,1)<=0.063
      bin9=[bin9; lengthes(j,1)];
   end
end
bin10=[];
for j=1:m
   if lengthes(j,1)>0.063 & lengthes(j,1)<=0.07
      bin10=[bin10; lengthes(j,1)];
   end
end
bin11=[];
for j=1:m
   if lengthes(j,1)>0.07 & lengthes(j,1)<=0.077
      bin11=[bin11; lengthes(j,1)];
   end
end
bin12=[];
for j=1:m
   if lengthes(j,1)>0.077 & lengthes(j,1)<=0.084
      bin12=[bin12; lengthes(j,1)];
   end
end
bin13=[];
for j=1:m
   if lengthes(j,1)>0.084 & lengthes(j,1)<=0.091
      bin13=[bin13; lengthes(j,1)];
   end
```

```
end
bin14=[];
for j=1:m
   if lengthes(j,1)>0.091 & lengthes(j,1)<=0.098
      bin14=[bin14; lengthes(j,1)];
   end
end
bin15=[];
for j=1:m
   if lengthes(j,1)>0.098 & lengthes(j,1)<=0.105
      bin15=[bin15; lengthes(j,1)];
   end
end
bin16=[];
for j=1:m
   if lengthes(j,1)>0.105 & lengthes(j,1)<=0.112
      bin16=[bin16; lengthes(j,1)];
   end
end
bin17=[];
for j=1:m
   if lengthes(j,1)>0.112 & lengthes(j,1)<=0.119
      bin17=[bin17; lengthes(j,1)];
   end
end
bin18=[];
for j=1:m
   if lengthes(j,1)>0.119 & lengthes(j,1)<=0.126
      bin18=[bin18; lengthes(j,1)];
   end
end
bin19=[];
for j=1:m
   if lengthes(j,1)>0.126 & lengthes(j,1)<=0.133
```

```
        bin19=[bin19; lengthes(j,1)];
    end
end
bin20=[];
for j=1:m
    if lengthes(j,1)>0.133 & lengthes(j,1)<=0.140
        bin20=[bin20; lengthes(j,1)];
    end
end
bin21=[];
for j=1:m
    if lengthes(j,1)>0.140 & lengthes(j,1)<=0.147
        bin21=[bin21; lengthes(j,1)];
    end
end
bin22=[];
for j=1:m
    if lengthes(j,1)>0.147 & lengthes(j,1)<=0.154
        bin22=[bin22; lengthes(j,1)];
    end
end
bin23=[];
for j=1:m
    if lengthes(j,1)>0.154 & lengthes(j,1)<=0.161
        bin23=[bin23; lengthes(j,1)];
    end
end
bin24=[];
for j=1:m
    if lengthes(j,1)>0.161 & lengthes(j,1)<=0.168
        bin24=[bin24; lengthes(j,1)];
    end
end
bin25=[];
```

```
for j=1:m
   if lengthes(j,1)>0.168 & lengthes(j,1)<=0.175
      bin24=[bin24; lengthes(j,1)];
   end
end
disp(['bsize1=', num2str(size(bin1,1))]);
disp(['bsize2=', num2str(size(bin2,1))]);
disp(['bsize3=', num2str(size(bin3,1))]);
disp(['bsize4=', num2str(size(bin4,1))]);
disp(['bsize5=', num2str(size(bin5,1))]);
disp(['bsize6=', num2str(size(bin6,1))]);
disp(['bsize7=', num2str(size(bin7,1))]);
disp(['bsize8=', num2str(size(bin8,1))]);
disp(['bsize9=', num2str(size(bin9,1))]);
disp(['bsize10=', num2str(size(bin10,1))]);
disp(['bsize11=', num2str(size(bin11,1))]);
disp(['bsize12=', num2str(size(bin12,1))]);
disp(['bsize13=', num2str(size(bin13,1))]);
disp(['bsize14=', num2str(size(bin14,1))]);
disp(['bsize15=', num2str(size(bin15,1))]);
disp(['bsize16=', num2str(size(bin16,1))]);
disp(['bsize17=', num2str(size(bin17,1))]);
disp(['bsize18=', num2str(size(bin18,1))]);
disp(['bsize19=', num2str(size(bin19,1))]);
disp(['bsize20=', num2str(size(bin20,1))]);
disp(['bsize21=', num2str(size(bin21,1))]);
disp(['bsize22=', num2str(size(bin22,1))]);
disp(['bsize23=', num2str(size(bin23,1))]);
disp(['bsize24=', num2str(size(bin24,1))]);
disp(['bsize25=', num2str(size(bin25,1))]);
disp(['m=', num2str(m)]);
disp(['min=', num2str(min(lengthes))]);
disp(['max=', num2str(max(lengthes))]);
disp(num2str(size(bin1,1)));
```

```
disp(num2str(size(bin2,1)));
disp(num2str(size(bin3,1)));
disp(num2str(size(bin4,1)));
disp(num2str(size(bin5,1)));
disp(num2str(size(bin6,1)));
disp(num2str(size(bin7,1)));
disp(num2str(size(bin8,1)));
disp(num2str(size(bin9,1)));
disp(num2str(size(bin10,1)));
disp(num2str(size(bin11,1)));
disp(num2str(size(bin12,1)));
disp(num2str(size(bin13,1)));
disp(num2str(size(bin14,1)));
disp(num2str(size(bin15,1)));
disp(num2str(size(bin16,1)));
disp(num2str(size(bin17,1)));
disp(num2str(size(bin18,1)));
disp(num2str(size(bin19,1)));
disp(num2str(size(bin20,1)));
disp(num2str(size(bin21,1)));
disp(num2str(size(bin22,1)));
disp(num2str(size(bin23,1)));
disp(num2str(size(bin24,1)));
disp(num2str(size(bin25,1)));
disp(num2str(m));
```