Karlsruhe Institute of Technology

Spatial distribution of pores in porous materials

(Räumliche Verteilung von Poren in porösen Materialien)

Master Thesis

of

Tianyu Lu

at the

Institute of Applied Materials
Microstructure Modelling and Simulation (IAM-MMS)
of the
Department of Mechanical Engineering
at the
Karlsruhe Institute of Technology (KIT)

Reviewer:	Prof. Dr. rer. nat. Britta Nestler
Supervisors:	Farshid Jamshidi, M.Sc. Willfried Kunz, M.Sc.

Duration: September 2021 – March 2022

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die hier vorgelegte Masterthesis selbstständig und ausschließlich unter Verwendung der angegebenen Literatur und sonstigen Hilfsmittel verfasst habe. Die Arbeit wurde in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde zur Erlangung eines akademischen Grades vorgelegt.

Karlsruhe, den 07. März 2022

Tianyu Lu

Acknowledgment

Here I want to thank Prof. Dr. rer. nat. Britta Nestler. She gives me the opportunity to do my master thesis at the Institut of Applied Materials Science. I would like to express the special gratitude to my supervisors Farshid Jamshidi and Willfred Kunz for their invaluable guidance, continuous support throughout the past few months. I would also like to thank Dr.-Ing. Patrcik Altschuh for his useful suggestions and technical helps.

Contents

Abstract

Porous materials have been developed rapidly in recent years and have many particular properties due to their special geometries, such as lightweight, large specific surface area, good energy absorption and high heat-transfer capacity. A commonly used parameter to characterize porous structures is the number of pores per inch (PPI). In this work, PPI is computed in three-dimensional (3D) space rather than in two dimensions, for instance from a two-dimensional image. The studied structures are computer-generated or reconstructed from computed tomography (CT) scan images of real structures. The images segmented by the marker-based watershed algorithm (a part of the pore network model) are also used in this work. All the computations are conducted using the solver Parallel Algorithm for Crystal Evolution in 3D (Pace3D) for counting the number of pores accurately. The algorithm for counting the number of pores is proposed and modified according to the distribution of pores in different cases.

This work is divided into three main stages: verifying the counting algorithm with an aligned structure, modifying it for complex structures, and applying it for real structures. The two modified algorithms are also validated by a periodic aligned structure, then two modified algorithms are used in complex structures to reduce the influence of periodic boundaries. The first modified algorithm is a subtraction of computed values and the second modified algorithm is an adjustment of the length of the measuring lines. The relationships between PPI and other important parameters, such as porosity, the total number of pores (Voronoi points) and strut radius, are observed. The effects of the stretching direction and degree of pores on PPI are also studied. The PPI of structures with different stretching factors in one or two directions is shown. In addition, the advantages and disadvantages of the two modified algorithms are also identified.

By reconstructing the real structure and computing the PPI of the 3D models, the PPI of the real foams is found to be smaller than the value given by the manufacturer. This difference is particularly evident in the structure with a large PPI. By observing the PPI in different directions, it is possible to find the stretching of pores.

1. Introduction

In this chapter, the traditional method for measuring PPI and the motivation to improve the traditional method are introduced. The purpose and structure of the thesis are also mentioned.

1.1. Motivation

With the advancement of scientific concepts and production technology, people have shown more concern about porous materials because of their special geometry, remarkable properties, and valuable functions in recent years. Different materials correspond to different applications. In the early 1990s, a working group for porous materials in the USA established a few areas as potential applications for porous materials in industrial production: porous materials are particularly attractive for lightweight construction, impact-energy absorption, acoustic damping, and heat-transfer applications [16].

Nowadays, people can gain more information about porous materials through gradually advanced characterization methods. A porous material morphology consists of the geometry that describes the pores' shapes and sizes as well as the structure of a porous medium. In addition, it includes the topology, which quantifies the way pores and throats are connected [33]. For metal foams, the size of cells is a particularly critical parameter, because many properties of foams are directly influenced by cell size. In the past few years, PPI (pores per inch) has been often used for characterizing metal foams. Different techniques of counting the number of pores over a standard length of one inch provide different methods for measuring PPI. The linear intercept method was primarily used when manual measurements were required to obtain data because it could be performed by drawing random lines on images of sections [34]. The traditional method is shown in Figure 1.1.

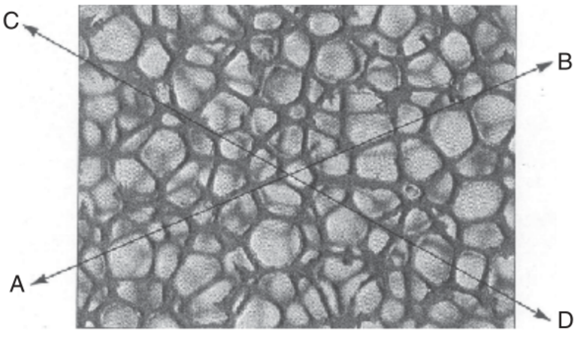


Figure 1.1.: Principle of the traditional method for PPI measurement [34].

However, the traditional PPI method has many limits. First, all pores cannot be clearly distinguished in a two-dimensional (2D) image. Second, porous materials are usually complex, so varying the locations and numbers of measuring lines result in different PPI numbers for a single porous medium. In addition, manual measurements cause heavy workload and subjective influences.

Computer calculation can solve the aforementioned problems. In computer-generated or computer-reconstructed structures, the pores are clearly distinguished by marker-based watershed algorithms and PPI can be computed in three different directions. As shown in Figure 1.2, with the counting algorithm, more measuring lines can be placed in the entire volume of the structure, which can eliminate the influence of location on the result and the randomness of manual measurements. With outstanding computing capacity, the calculation can take less time and provide more accurate information about porous materials.

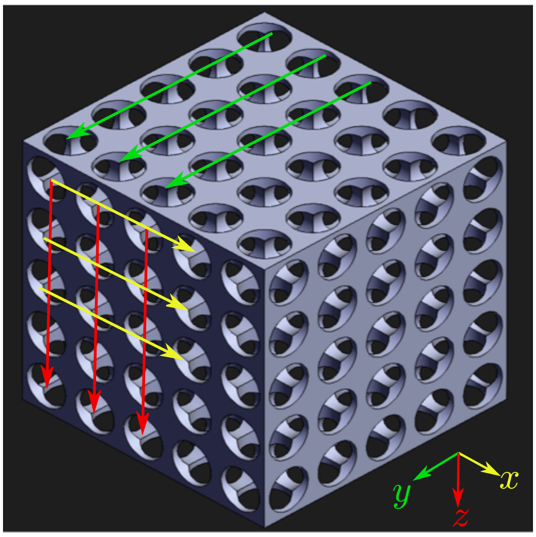


Figure 1.2.: Principle of the computer calculation method for PPI measurement[1].

1.2. Goal of the thesis

The first objective of this work is to implement a counting algorithm that can accurately compute the number of pores in a porous structure. By knowing the precise number of pores, it is possible to gain the accurate PPI (pores per inch). The algorithm must be applied in three dimensions, which demonstrates the superiority of a computer calculation over a traditional PPI measurement. At the same time, for further analyses and characterization, many topological and morphological properties of the structure should be taken into account. Therefore, a model in which individual pores can be accurately distinguished is crucial.

The second objective is that once an accurate PPI value for a porous structure is obtained, the other tools (mentioned in chapter 3) can be used to measure other basic parameters, such as porosity, strut radius, to determine the relationship between PPI value and other

[1] Source: `http://grabcad.com/library/8mm-cell-foam-1.`, access data: 01-20-2022

basic parameters. This can make PPI values more informative and make PPI be reused to characterize porous structures. For real structures, it needs to be determined whether the PPI of the reconstructed structure is the same as that given by the manufacturer, and if not, how large the difference is.

The long-term goal is to use more suitable solutions for computing the PPI of different structures because the structure of porous materials is often complex. To maintain a high level of reliability of the measurement results for real structure, the validations for aligned structures must also be taken into account.

1.3. Structure of the thesis

This section is intended to give a brief overview of the contents found in the chapters constituting this thesis. Chapter 2 is related to the state of research for porous materials. The construction of their network models and a few classic parameters are introduced. Chapter 3 starts with preprocessing tools and their algorithm fundamentals used in this study. According to these algorithms, suitable segmented images are achievable for this study.

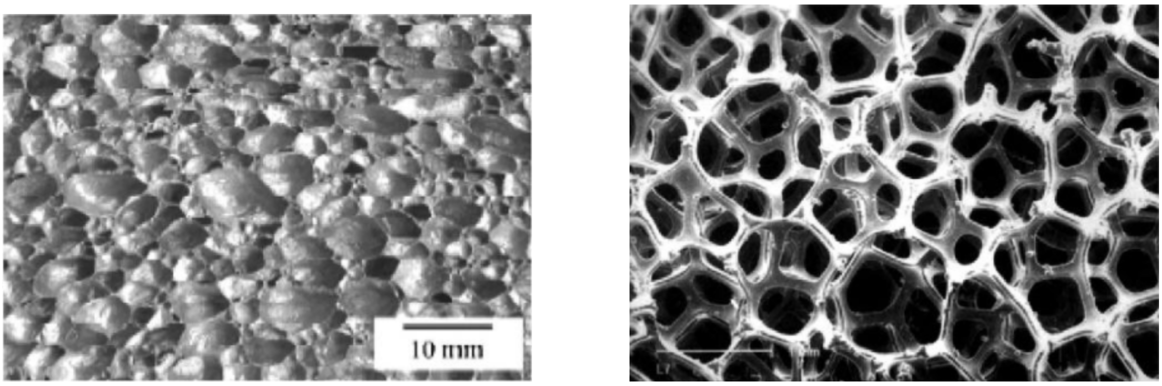
Chapter 4 gives the method of computing PPI as a very important parameter for porous materials. Depending on the type of porous media, different algorithms are required. Moreover, the studied structures are also introduced, for example, computer-generated structures, computer-reconstructed structures. The results and analyses are shown in chapter 5, the relationships and the variation trend between different parameters and PPI are described. Chapter 6 summarizes the main findings of this work and presents questions met during the study and have the potential to be solved in the future.

2. State of research

This chapter gives the basic knowledge about porous materials. Besides, the pore network modeling of porous materials and the way how to construct the model are introduced. Pore network models are valuable and useful tools for understanding and predicting mesoscale phenomena. The segmentation of binary images used in the thesis is the basis of pore network modeling. By using them, results are obtained more quickly with less computational work [35].

2.1. Porous materials

What are porous materials? So-called materials must possess two essential characteristics: one is that the material contains a lot of pores, and the other is that the pores are designed specifically to achieve the expectant index of the material's performance [29]. According to the structure, porous materials can be classified into two types: closed-cell and open-cell. For example, a honeycomb material, in which the structure looks similar to a two-dimensional (2D) array of the hexagon and the pores are isolated: this type of material has closed pores and the cross-sectional shape of pores can also be circular, triangle, etc. By open-cell structures, on the other hand, the solid part of a material is a three-dimensional (3D) reticulated structure and the pores are connective. Figure 2.1 shows closed-cell and open-cell foamed materials. In general, the materials with low or middle porosities have closed pores and the high porosity materials exhibit more complex morphologies, which means that both cases happen possibly on high porosity materials.



(a)

(b)

Figure 2.1.: Two different types of porous structure: (a) a closed-cell bubblelike foamed material of aluminum foam; (b) an open-cell foamed material of nickel foam [29].

2.2. The network model of porous structure

As Figure 2.2 shows, both closed-cell and open-cell structures consist of irregular polyhedral chambers and a three-dimensional strut network that surrounds them. The difference is that the cells of an open-cell structure are connected via open face, while the cells of closed-cell are isolated, between two cells is solid face.

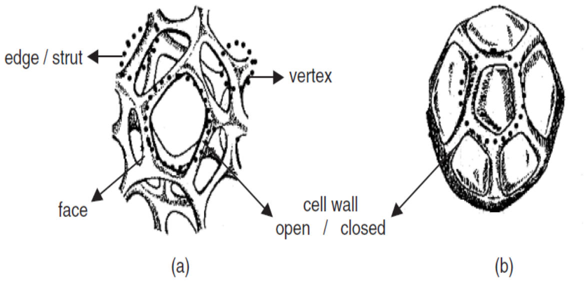


Figure 2.2.: Components of a cell unit in (a) open and (b) closed cell foams [34].

The computer-generated or computer-reconstructed structures used in this work are open-cell, so the pore network modeling is used to describe complex morphology in structure with regular shapes. The current pore network model (PNM) is mainly composed of two parts: pores and throats. The throat represents the elongated space where the pores are linked. The pore network model is an idealization and simplification of a variable porous structure using regular shapes such as spheres and columns. The spatial location and geometric parameters of these spheres or columns are key to the construction of the model.

2.2.1. Conventional pore network models

PNMs have evolved from simple to complex, and there are many ways to construct a PNM, for example [35]: first, statistical reconstruction, the basic morphologic parameters are used to create a statistically equivalent network. Such as the work of Rezanezhad et al. [30], the information from 2D images can be used to provide 3D, high-resolution images and the analysis of 3D images describes the distribution and size of pores and their effects on the flow. The second, direct mapping model, is to map a network structure directly onto a porous void space. The common approaches include the medial axis algorithm, which has been studied and improved by Lindquist et al. [28] and Liang et al. [27] since the 1990s; maximal ball algorithm, which has been refined by Dong et al. [12], they proposed a clustering algorithm that can more clearly define pores and throats by making family trees for the maximal balls according to their size and rank.

In this work, a new graph-based model has been mentioned: the PNM is defined as a graph whose nodes represent the pores and whose edges represent the connections between the pores, so-called throats. Thus, the distribution of pores and the connections between pores can be clearly presented. A new algorithm called subnetwork of the over-segmented watershed (SNOW), which was based on the marker-based watershed algorithm and developed by Gostick et al. [17], will also be applied to avoid over-segmentation. The watershed algorithm assigns each pixel or voxel to a region, and if a large number of small regions appear, this is known as "over-segmentation" [8].

2.2.2. Extraction of pore network models

This subsection describes the specific extraction process of several mentioned models. For instance, medial axis algorithm, maximal balls algorithm and SNOW algorithm.

Medial axis algorithm

This approach converts the pore space images into a medial axis, which is a simplified representation of the pore space and acts as a topological skeleton. Baldwin et al. [6] presented a thinning algorithm to extract a medial axis. As Figure 2.3 shows, starting from the voxels closest to the solid matrix, the voxels in pore space are eroded step by step in a certain direction and at a certain speed. The erosion sequence of the voxels is marked and the last eroded voxel should be on the medial axis. It can be seen that the medial axis of part a~h are yellow voxels, and the medial axis of i and j are red voxels. The position with the shortest distance from the medial axis to the solid phase namely the position, where the medial axis marked by the smallest number in Figure 2.3, is considered to be the pore throat. Based on this information, a pore network model can be constructed.

The medial axis algorithm can easily obtain the interconnectivity of the pores, however, it is still difficult to identify individual pores. The irregularities in pore space and the choice thinning (erosion) directions can lead to the existence of multiple medial axes in the same pore, the high likelihood for a big pore to be divided into several small pores and throats. Therefore, various merging algorithms need to be developed to trim the skeleton and improve the partition of pores [19]. In conclusion, this algorithm presents the interconnectivity well, but is not suitable for this thesis because pores in this work must be better identified and high-porosity structures should not be over-segmented.

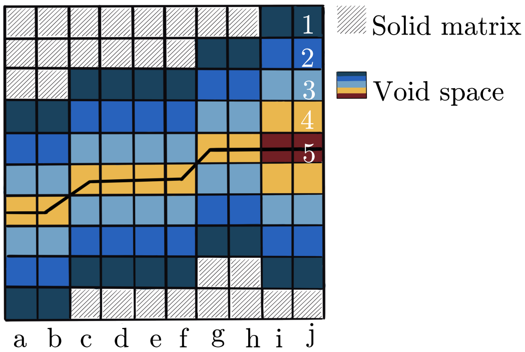


Figure 2.3.: Illustration of the medial axis algorithm, adapted from Fig. 3 in Ref [25].

Maximal balls algorithm

The maximal ball (MB) algorithm was developed by Al-Kharusi and Blunt [1]. Dong et al. [12] improved their algorithm and found a clustering algorithm that classifies maximal balls into pores and throats. The MB algorithm takes any voxel in the pore space as a reference voxel and tries to find the largest inscribed spheres centered on each voxel that just touch the grain or the boundary. As Figure 2.4 shows, the red voxel is an initial point, extends in 8 directions until it encounters a solid voxel or boundary (voxels marked 4 in the diagram). The distance from the red voxel 1 to the light blue voxel 3 is regarded as the lower limit of the radius of this inscribed sphere, and the distance from the red 1 voxel to the blue 4 voxel is regarded as the upper limit of the radius, the true radius of this maximum sphere lies between these two radii. Once all spheres are found, those spheres, which are included in other spheres, are considered as redundant and removed, while the rest are called maximal balls. If all maximal balls have been found, the next step is to divide them into pores and throats. First, all MBs are sorted by size, from largest to smallest, each group contains MBs of the same size. Next, starting from the maximal ball with the largest radius and it is defined as a pore, all its smaller or same sized overlapping neighbors are absorbed by it and ranked the second generation, so the ball is a parent in this cluster [12]. When two families have a common child as in Figure 2.5, the common child defines a throat and a pair of pore throat chains can be constructed. The black balls are on the chain along with the skeleton, the white balls are not on the skeleton but can describe pore space. But this algorithm requires a large amount of computation and it is also just suitable for low or middle porosity structures [11].

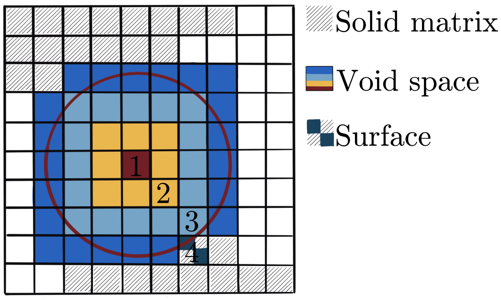


Figure 2.4.: Illustration of the maximal balls algorithm, adapted from Fig. 4 in Ref [25].

SNOW algorithm

In this thesis, a subnetwork of the over-segmented watershed (SNOW) algorithms, which was developed by Gostick et al. [17], is used to extract the PNMs accurately. Section 3.3 provides more details about the algorithm and its implementation in a tool.

2.3. Important parameters

In this section, some basic and important parameters of porous material are introduced. The physical behavior of porous materials depends on their microscopic structures, so some

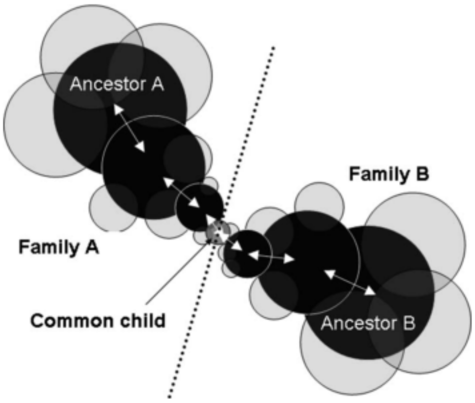


Figure 2.5.: Two families A and B with a common child [12].

parameters, such as porosity, pore size and shapes are crucial. Some of these are variables that need to be controlled in this work by computer generation.

2.3.1. Porosity

Porous materials can be classified by porosity: low (<30%), middle (30~60%) and high porosity (>60%). The porosity ε can be defined as the fraction of void space (pore) V_v relative to the apparent total volume V_t [22]

$$\varepsilon = V_\mathrm{v}/V_\mathrm{t}\,. \tag{2.1}$$

There are two types of porosity for porous structures: one is the "open porosity" ε_o, which is due to the volume of voids surrounded by foam strut network and led by open pores; another is the "closed porosity" ε_c, which is led by closed pores. The sum of them is the total porosity. For open-cell structures studied in this thesis, the open porosity is taken as the total porosity.

2.3.2. Pore size and shape

Figure 2.6 illustrates the essential geometric characteristics and shapes of porous structures:

- d_pore : pore diameter
- d_face : diameter of windows
- d_edge : thickness of ligament
- l_edge : length of ligament

Beer et al. [7] proposed a method to determine the essential geometric characteristics of the metal foam structures. Firstly, they used photography devices to get the images. Next, they utilized an analytical imaging software to choose a suitable part of the structure to measure the geometric parameters, such as Fiji-ImageJ 1.53c. Finally, they used the Kline-McClintock's

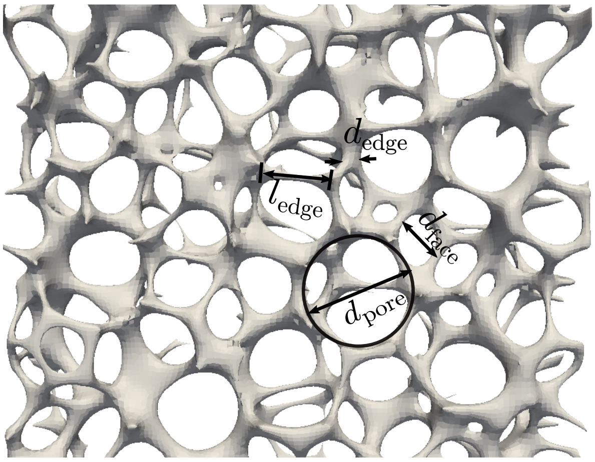


Figure 2.6.: Illustration of geometric parameters of open-cell structure.

method [21] to perform the uncertainty of the measurement process. The correct identification and quantification of pore size and shapes are crucial for the development of porous materials.

2.3.3. Specific surface area

The "specific surface area" is defined as the detectable area of solid surface per unit mass or volume of materials [31]. The specific surface area is an important property of foam structures, which is particularly relevant for heat and mass transform, adsorption, heterogeneous catalysis, and reactions on surfaces. Different methods or experimental conditions deliver different values of the specific surface area. For example, Bucimuman et al. [10] used a theoretical correlation for estimating the specific surface area (Equation 2.2), and the used open-cell structure is described as a theoretical model of the tetrakaidecahedron

$$S_V = C \cdot \frac{1}{d_{\text{edge}} + d_{\text{windows}}} \cdot (1 - \varepsilon)^n \,. \tag{2.2}$$

In this equation, S_V is the volumetric specific surface area; d_{edge} and d_{windows} represent the diameter of strut and window in open-cell structure, respectively, and ε is the porosity; The two constants C and n can be obtained by a geometrical approach.

2.3.4. Pore density: Pores Per Inch (PPI)

Previously, PPI is used as an approximate commercial characteristic designated by the manufacturer. Over recent years, PPI is usually just used to classify rather than to characterize open-pore structures, because the ratio of pore size and ligament thickness may vary in two structures with the same PPI. Therefore, many people believe that PPI value cannot give effective information. In this thesis, PPI is associated with other parameters, the relationships between them can make PPI more informative.

3. Preprocessing and theoretical background

The goal of this work is to count the PPI number of porous structures by using computer-aided methods. In the first step, computer-generated or computer-reconstructed structures are applied, which have high similarities with real structures. And then diverse tools are used to do some preprocessing, which makes it possible to establish satisfying PNM. A structure with well-identified pores is the basic of counting and calculation. Throughout this work, all the numerical simulations are conducted using the solver Parallel Algorithm for Crystal Evolution in 3D (PACE3D), which is developed at Karlsruhe University of Applied Sciences, Germany, to study the evolution of free boundary problems with the phase-field method for various phenomena and especially multi-physics applications [18]. In order to combine hundreds of computed tomography images and transform them into a real structure, software Fiji-ImageJ 1.53c is also be applied to process images.

3.1. The tool: *generatemembrane*

The tool *generatemembrane* in PACE3D is for generating membrane-like structures and the algorithm is developed by Altschuh et al [3]. The main generating process consists of the following three steps:

(i) Spheres filling (Voronoi-points setting)

In the first step, spheres should be placed in the defined domain so that they can fill the 3D space as compact as possible. The number of the spheres is related to the amounts of Voronoi points. The first sphere is placed randomly in the domain. The second sphere is also set in a random place with a different location as the first sphere and then moves to the first sphere [4]. The third-placed sphere cannot intersect with or on the same straight line with the other two spheres and move towards the previous two spheres, after each iteration, the sphere is one step closer to an ideal location. Each following sphere is placed in the same way until all of them occupy an optimal location.

(ii) Construction of Voronoi-diagram

A Voronoi diagram is created by decomposing a 3D domain into Voronoi-regions according to the nearest-neighbor rule and the regions are related to the center of spheres. The nearest-neighbor means that each point is associated with the region closest to it [5]. In other words, a Voronoi-region is composed of the points whose distance to its center is closest than to the center of other regions. Figure 3.1 shows a 2D Voronoi diagram.

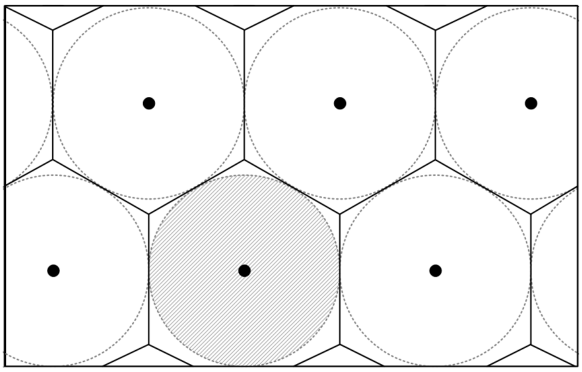


Figure 3.1.: A 2D Voronoi-diagram with hexagonal regions [32].

(iii) Strut generation

To form the strut of cellular structure, densely packed spheres with defined uniform radii are placed on the boundary of Voronoi-regions, where at least three regions are connected. The thickness of the strut can be controlled by the radii of packed spheres and have also direct effects on pore size. Figure 3.2 shows the process of construction of struts.

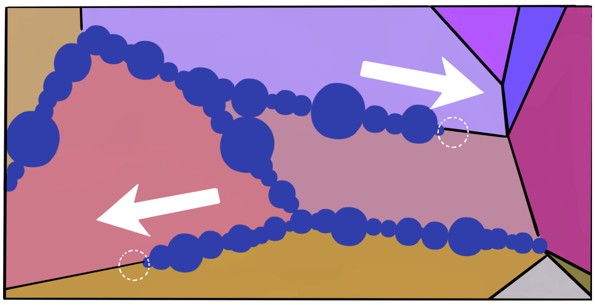


Figure 3.2.: The construction of struts along the edges of Voronoi-regions [3].

The tool *generatemembrane* can adjust the properties of generated structures via parameters. The following is a brief description of the parameters used in this work:

1. *-c*: with - - *count*, the number of Voronoi-points, which are placed in the domain, can be set.

2. *-b*: with - - *centroidal-voronoi*, a centroidal Voronoi tessellation is used, in which the Voronoi generators are the centroids of the corresponding Voronoi regions.

3. *-f*: with - - *fit-to-porosity*, porosity of the structure is set.

4. *-w*: with - - *stanchion-radius*, the radius of stanchion of the structure is set.

5. *-s*: with - - *seed*, the seed for the pseudo-random generator is set, which makes it possible to reproduce results of the generation.

6. *-j*: with - - *x-stretching*, the amount of stretching applied along the x-axis.

7. *-k*: with - - *y-stretching*, the amount of stretching applied along the y-axis.

3.2. The tool: *skeletonize*

In this work, the Euclidean distance field (mentioned in section 3.3) can be provided by the tool skeletonize. More details about the solving of the Euclidean distance field can be found in [2]. The shortest Euclidean distance $d(\mathbf{x}, \mathbf{s})$, for any voxel to the nearest voxel on structure surface, can be expressed in terms of a signed distance function $D(\mathbf{x})$. $\mathbf{x}$ is any voxel in domain Ω, s is the voxel on surface $\partial_{\mathrm{s}}\Omega$ and the domain Ω is a part of Euclidean space $\Omega \subseteq \mathbb{R}^3$.The shortest Euclidean distance is

$$d(\mathbf{x}, \mathbf{b}) = |\mathbf{x} - \mathbf{b}|_3 = \sqrt{(x_1 - b_1)^2 + (x_2 - b_2)^2 + (x_3 - b_3)^2}\,. \tag{3.1}$$

The signed distance function $D(\mathbf{x})$ is

$$D(\mathbf{x}) = d(\mathbf{x}, \mathbf{s})\mathrm{sgn}(f(\mathbf{x}))\,, \tag{3.2}$$

where $f(\mathbf{x})$ is an initialisation function and the function $sgn(f(\mathbf{x}))$ is used to determine whether the point $\mathbf{x}$ is in pore space, structure, or on the surface; the function $sgn(f(\mathbf{x}))$ is defined as:

$$\mathrm{sgn}(f(\mathbf{x})) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega^+ \\ -1 & \text{if } \mathbf{x} \in \Omega^- \\ 0 & \text{if } \mathbf{x} \in \partial_{\mathrm{s}}\Omega \end{cases}, \tag{3.3}$$

where Ω^+ represents the pore space; Ω^- represents the structure; $\partial_{\mathrm{s}}\Omega$ represents the surface.

In the tool *skeletonize*, thinning algorithm is applied to the extract medial axis, which can just deliver the topological properties and connectivity, but not the information about distance. So, the signed distance function is combined with thinning algorithm, that the tool *skeletonize* is used to get the pore size

$$r(\mathbf{x_s}) = D(\mathbf{x_s})\,, \tag{3.4}$$

where $\mathbf{x_s}$ is a voxel on the skeleton (medial axis).

3.3. The tool: *extractnetwork*

The algorithm for the tool *extractnetwotk* is a marker-based watershed algorithm and is divided into two stages: segmentation of the pore space and extraction of the PNM based on the segmented pore space. In this work, only the first stage of *extractnetwork* is used, the segmentation of pore space can directly influence the PPI.

The watershed transform is a complex morphological technique, that the signals of images are used to segment the images. For a grayscale image, the pixel values are the signals that can be applied. As Figure 3.3 shows, the pixel values are used to identify peaks and basins (local maximum and local minimum). Flooding begins at the lowest points of basins (local minimum) until the waters from different basins meet each other and the segmentation is completed. The watershed lines (boundaries) consist of meeting points of waters. The number of markers also affects the segmentation if local minima as markers without filtering are used. This is likely to lead to over-segmentation of an image.

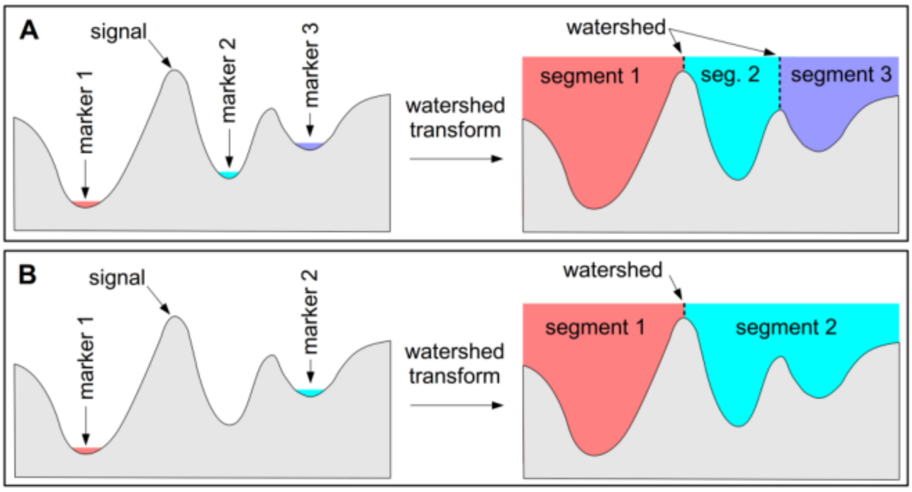


Figure 3.3.: Examples of the marker-based watershed transform with different markers: (A) three local minima act as markers (B) two markers are selected, the water in segment 2 flows into the neighbouring basin and forms a boundary with segment 1, the number of segments is the same as the number of markers [15].

If a binary image is used, the Euclidean distance field will be combined with the marker-based watershed algorithm. Subsequently, the first stage of the extraction algorithm is divided into the following five conceptual steps [17]:

1. **Preparation of the distance field**
   In the first step, the image is prepared for segmentation by calculating its Euclidean distance field, then smoothing it with a Gaussian filter with variance $\sigma= 0.35$ to remove spurious peaks, which are misidentified in the distance field.

2. **Identification of peaks (local maxima) in the distance field**
   A maximum filter with a spherical structuring element of radius $R = 5$ is used to determine the maxima in the distance field. The R is difficult to define, when it is too small, spurious maxima are found, if R is too large, some maxima disappear.

3. **Elimination of peaks on saddles and plateau**
   After the previous steps, the initial set of local maxima is available. On saddles and plateau, a few markers appear that may cause over-segmentation. Therefore, the Superfluous and repetitive markers should be eliminated.

4. **Merging neighbouring peaks**
   If two markers are close to each other, they cannot form a saddle or plateau. As Figure 3.4 shows, if centered spheres with the corresponding distance values as radius are placed on the considered markers and they overlap each other significantly, that means the set of overlapping markers should be reduced to one marker, all markers satisfy the following formula.

$$d(m,n) < \min\{D(m), D(n)\}; (m,n) \in M \times M, m \neq n, \tag{3.5}$$

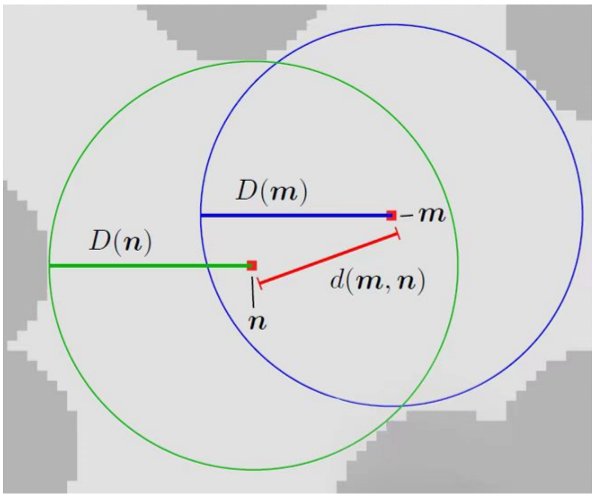


Figure 3.4.: An example of merging neighbouring markers [24].

where m and n are two nearby markers, $d(m,n)$ is the Euclidean distance between the two markers, $D(m)$ and $D(n)$ are distances between markers and the nearest solid surface. When the Euclidean distance between two markers is still less than their distances to the surface, the markers with the smaller distance to the surface should be eliminated. If the two markers have the same distance to the surface, (randomly) should be removed.

5. **Segmentation of pore space**
   After completing the above steps, an accurate set of markers is gained and the marker-based watershed algorithm is now used to segment the pore space.

The second stage is based on the first stage. The goal of the second stage is to obtain information from the watershed-segmented images to construct a pore network model. The important information includes connectivity, pore and throat geometric properties. Figure 3.5 shows a segmented-structure by *extractnetwork*, all pores are clearly identified. This useful pore network model provides an advantage for calculating PPI.

3.4. ImageJ

Fiji-ImageJ is an open-source software that is used in this work to process the images. At the later stage of this work, ImageJ is useful to study real structures. The specimens are provided by IAM-MMS, with the help of computed Tomography, information of the internal structure of specimens can be obtained. Therefore, a stack of CT images is observed to reconstruct the real structure, and image processing seems to be particularly important.

It can be seen in Figure 3.6 that the original CT images contain the metal in the outer layer, and the contrast between pores and solid parts is not very clear. The outer non-foam structure has to be removed and a suitable threshold must be chosen to ensure that the proportion of

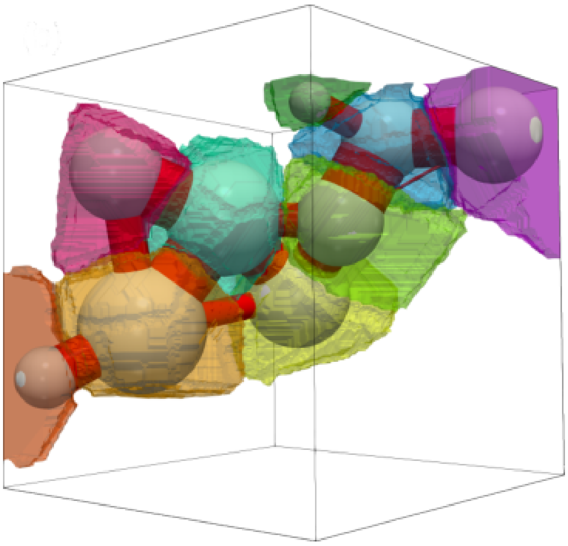
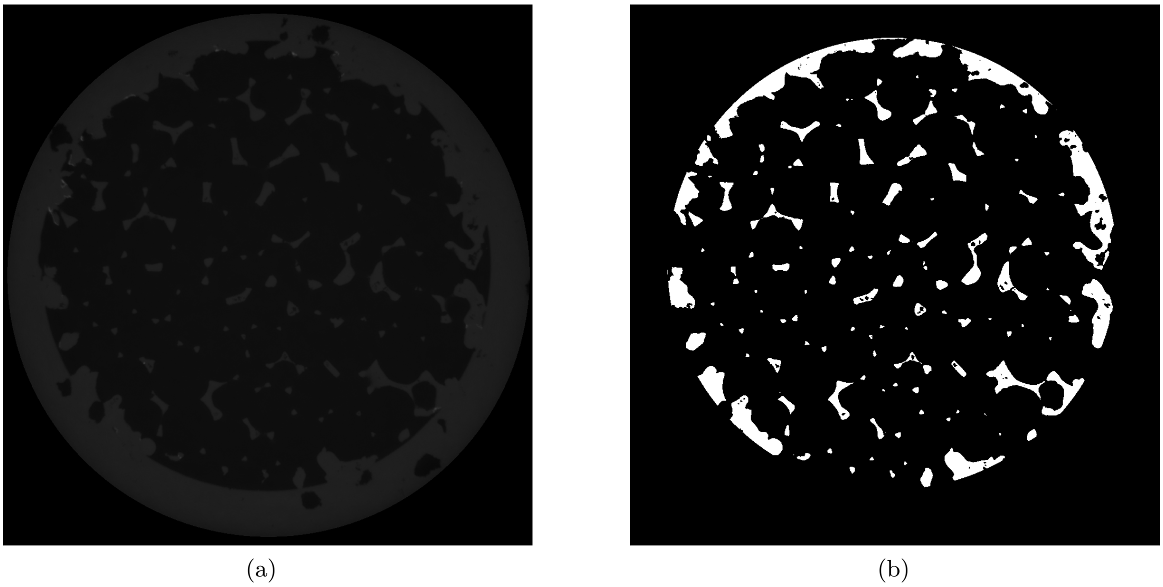


Figure 3.5.: A 3D representation of a structure that the pore space has been segmented by tool *extractnetwork*, the segments are marked with different colors respectively [24].

pores in the images is close enough to reality. The detailed processing steps and results are described in chapter 4.



(a)

(b)

Figure 3.6.: An example of image processing: (a) a CT image without processing; (b) the same image after processing.

4. PPI computation methods

In this chapter, the entire workflow of computing PPI is mentioned. The use of simple and complex structures, the choice of parameters, and a validation case study are also included. The algorithm for computing PPI has also undergone several adjustments, due to the complexity of the pore distribution and specifical boundaries of generated structures. The original algorithm is modified by two approaches. This chapter also describes the encountered problems and the solutions proposed for them.

4.1. The computation of PPI number

Figure 4.1 shows the entire workflow of the PPI computation method studied in this thesis. The structures used in this thesis are transformed into stacks of binary images, structure space is 1, pore space is 0. The Euclidian distance field (mentioned in section 3.2) is the basis of the segmentation for binary images, consequently, the tool *skeletonize* is used to calculate the Euclidian distance field of the binary images, structure spaces have negative values, pore spaces have positive values. Next step, the tool *extractnetwork* (mentioned in section 3.3) can use the binary images and their distance fields with the marker-based watershed algorithm to segment the images. The PPI-counting algorithms can be used in segmented images to obtain an accurate number of PPI.

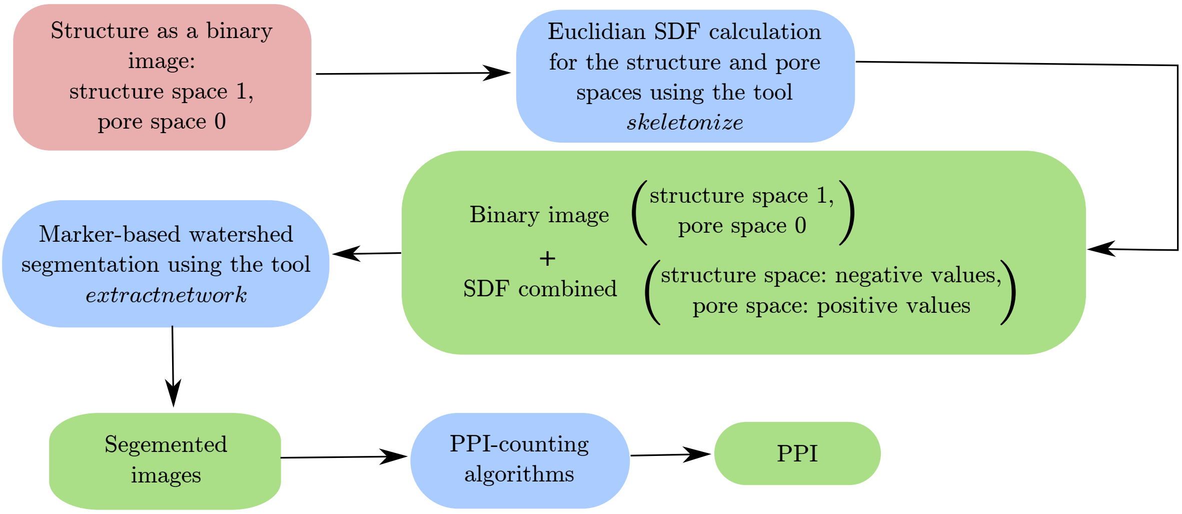

Structure as a binary image:
structure space 1,
pore space 0
Euclidian SDF calculation for the structure and pore spaces using the tool *skeletonize*
Binary image (structure space 1, pore space 0)
+
SDF combined (structure space: negative values, pore space: positive values)
Marker-based watershed segmentation using the tool *extractnetwork*
Segemented images
PPI-counting algorithms
PPI

Figure 4.1.: The workflow of the computation of the PPI number.

4.2. The original logic of algorithm

Things go from easy to hard, the algorithm verification is no exception. Using an appropriate PPI counting algorithm, how many pores exist in the entire domain should be ascertained. The aim of the algorithm is to determine the PPI number using straight lines across the porous medium in three directions. These lines are called measuring lines, some pores may not be identified, when the number of measuring lines is insufficient. The original logic of the PPI counting algorithm is simple: counting the number of pores that each measuring line passes through. This should be done for all measuring lines in the entire domain. Dividing the total number of pores by the total number of measuring lines gives the (average) PPI number.

It is worth mentioning that if a measuring line does not meet any pore, in other words, the measuring line passes only through the skeleton, the achieved number of counting pores is zero. These lines are excluded when calculating the average PPI number, otherwise, they reduce the average and lead to inaccuracies. In the segmented images obtained from *extractnetwork* (chapter 3), each pore is assigned a different index to provide an easier method for counting the number of pores. Figure 4.2 shows a schematic representation of counting pores using the PPI counting algorithm.

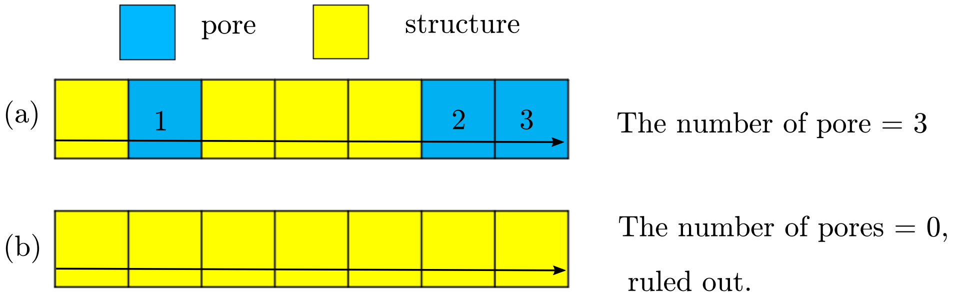
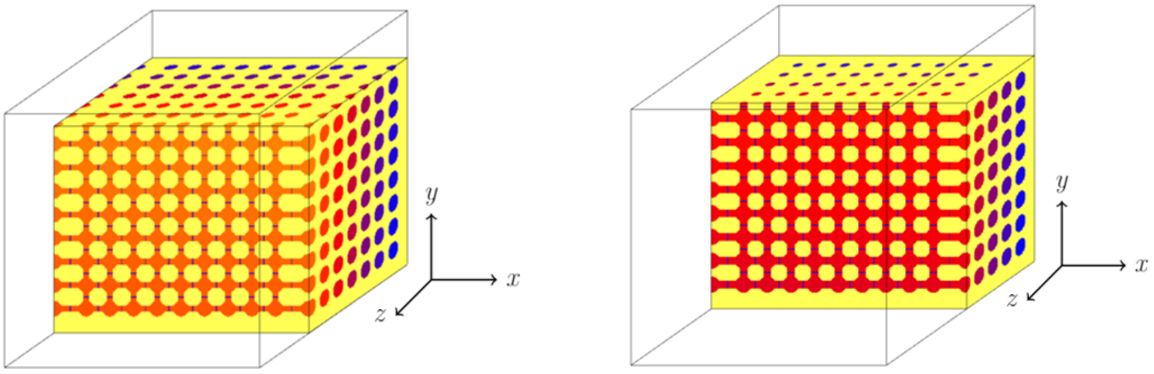


Figure 4.2.: The original logic for the counting algorithm: (a) the normal measuring line which passes through three pores; (b) the specific measuring line, which does not face any pore and should be excluded.

4.2.1. Validation of original algorithm

An aligned (highly ordered and uniform) structure is generated to validate the original algorithm, whose number of pores is known, namely 891 pores in the whole domain; 11 pores are placed in the x-direction, 9 pores are in the y and z-direction; The pores in the y-z plane at the boundary of the structure, namely at $x = 0$ and $x = 302$ (the minimum and maximum of the domain) are not interconnected.

The results are shown in Table 4.1, due to the changes in interconnectivity on the boundary, the x-direction result has a slight deviation from the set value, but this difference is to be expected. The reason is, when using the marker-based watershed algorithm, the connections between the pores, i.e., the pore throats, are also divided as the part of the pores. The numbers of pores in the other direction are the same as the setting. A preliminary conclusion can be drawn that this algorithm works for counting the pores in aligned structures.



(a)

(b)

Figure 4.3.: 3D presentation of an aligned structure: (a) the pores on the boundary can be detected; (b) because there are no pore throats, at a specific height, the pores on the boundary cannot be detected. The yellow part represents the solid structure.

Table 4.1.: The measured numbers of pores of the designed aligned structure.

Direction	x-axis	y-axis	z-axis	Average
Number of pores	10.465	9	9	9.488

4.3. The modified algorithms

The structures generated by *generatemembrane* are periodic. In each step, the domain is considered to be boundless; if there are N slices in one direction, the first slice is identical to slice N, so the pores that appear on these two slices represent the same pore. For periodic structures, the problem of double-counting of pores has emerged. To reduce the influence of the periodic boundary, the original algorithm should be modified.

Two possible options are tried: the first one, the measured number of pores can be adjusted. When the pores are detected both at the beginning and the end of the measuring line, because the structure is periodic, these two pores represent one single pore. As shown in Figure 4.4a, the measured value needs to be subtracted by one to exclude double-counted pores. The second option is that the measured value will not be changed, the measuring range will be adjusted. When the pores are detected both at the beginning and the end of the measuring line, the measurement range starts at the second pore and ends at the penultimate pore, as shown in Figure 4.4b. To show the modified algorithms more clearly, Figure 4.5 are the 3D presentations of the modified algorithms. The results obtained by these two algorithms need to be compared and their respective advantages or weaknesses should be analyzed. The comparison is described in chapter 5.

In addition, it is noteworthy that the length of measuring lines in the original algorithm and the first modification is the same as the length of the domain. In the second modification, the length of measuring lines is no longer invariant because they pass through different pores and the starting and ending points of the lines are also variable. The PPI can no longer be calculated by directly defining the entire domain as one inch, so in this work, a new concept is

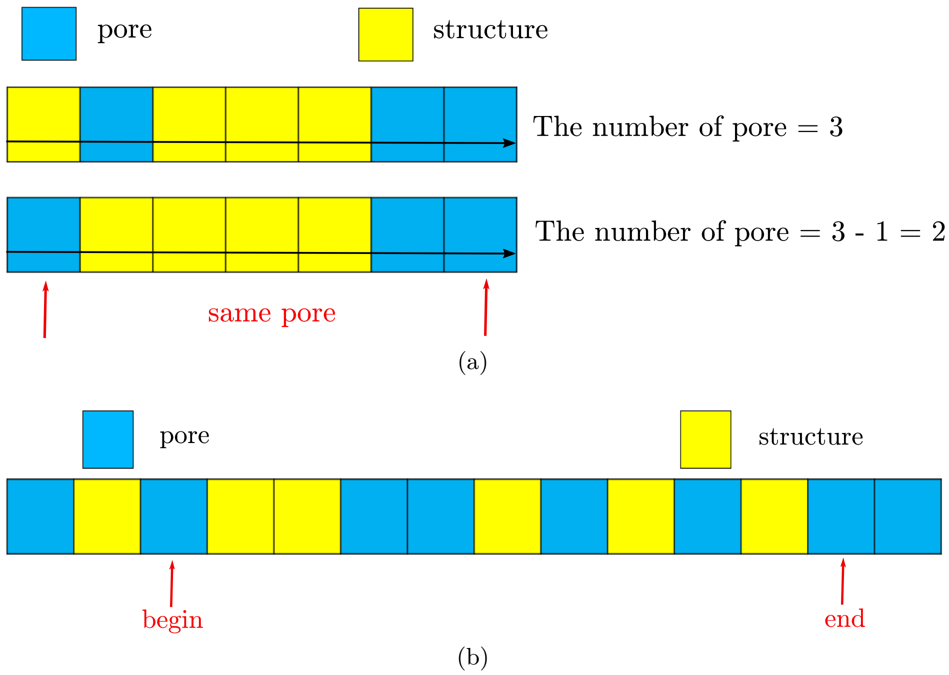


Figure 4.4.: The modified algorithms: (a) the first modification: subtraction of measured values; (b) the second modification: change of the measuring range.

used to represent PPI: PP100C. PP100C is pores per 100 voxel cells, which can be written as

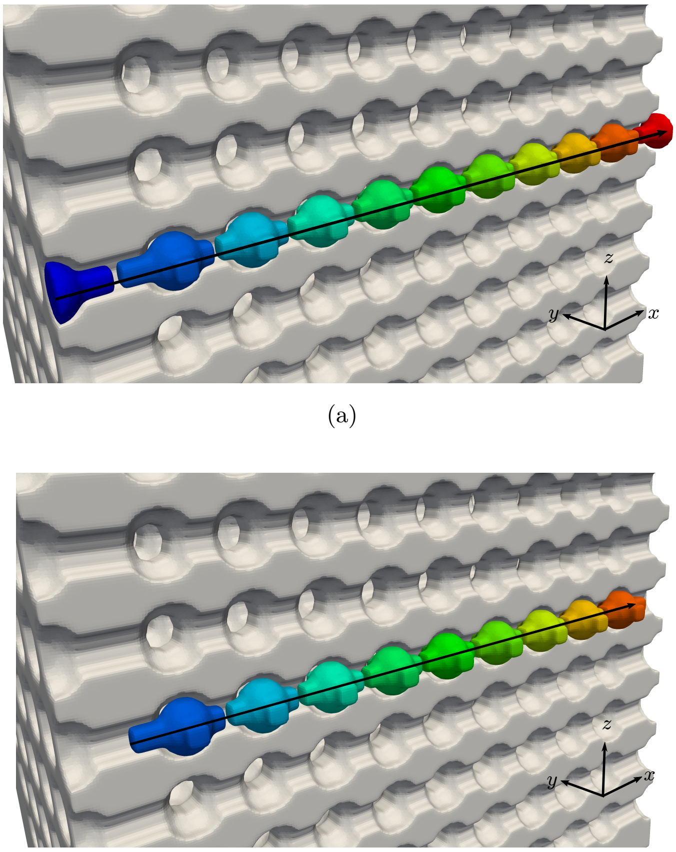
$$\text{PP100C} = \frac{C_\text{t}}{l_\text{m}} \cdot 100\,, \tag{4.1}$$

where C_t is the measured number of pores for a measuring line, l_m the length of the measuring line, unit is a cell. With PP100C, it is easier to convert between measured values and PPI, e.g., if 100 cells are assumed to be one inch, the PP100C is equal to PPI, if 200 cells are assumed to be one inch, in this case, the PPI is twice the PP100C. In chapter 5, all results are represented by PP100C instead. The concept of PP100C is to standardize the values obtained by each measuring line and make it more intuitive.

4.3.1. Validation of the modified algorithms

A periodic aligned structure is used to validate two modified algorithms. The domain of the structure is 672 cells, 23 full spheres and 2 half-spheres are arranged in all directions and the half-spheres on the boundaries are also divided into pores. Because of the periodicity, the first and last pores along measuring lines are considered to be the same pore, subsequently, the computed number of pores is expected to be close to 24. Figure 4.6 shows the periodic aligned structure. In Table 4.2 the computed results are expressed in PP100C and the PP100C can be converted into a number of pores by the following formula

$$N_\text{p} = \frac{\text{PP100C} \cdot \text{domain}}{100}\,, \tag{4.2}$$



(a)

(b)

Figure 4.5.: The 3D presentation of two modified algorithms: (a) the first modified algorithm; (b) the modified second algoritm. The colored part represents the pores and the white part represents the solid structure.

where N_p represents the number of pores. The relative difference between the two modified algorithms can be written as

$$\text{The relative difference} = \frac{\text{PP100C}_\text{second} - \text{PP100C}_\text{second}}{\text{PP100C}_\text{second}} \, . \tag{4.3}$$

By using Equation 4.2, the number of pores computed by the first modified algorithm is 23.7 and the number of pores computed by the second modified algorithm is 24.19, both results are very close to 24. The relative difference between the two modifications shown in Table 4.2 is also small. It can be concluded that both modified algorithms work well in the periodic aligned structure.

4.4. Computer-generated structures

As described in chapter 3, the tool *generatemembrane* can be used to generate complex structures. In this thesis, two types of structures are studied: the porosity is given; the strut

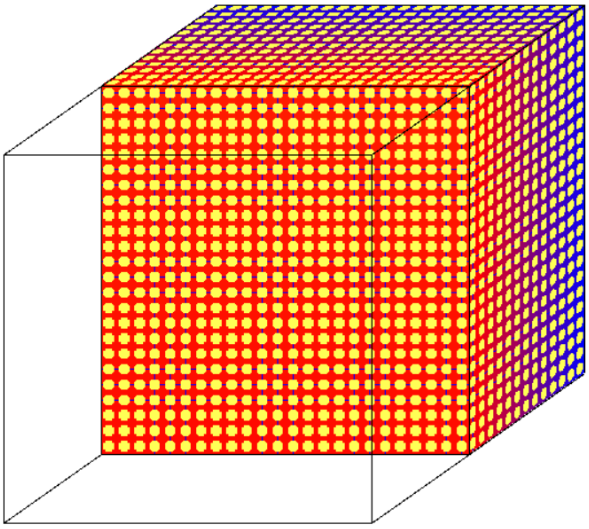


Figure 4.6.: The 3D presentation of the periodic aligned structure (yellow part represents solid).

Table 4.2.: The computed PP100C for the periodic aligned structure.

The modified algorithms	The computed PP100C for the periodic aligned structure in different directions		
	x-direction	y-direction	z-direction
The first modification	3.53	3.53	3.53
The second modification	3.60	3.60	3.60
The relative difference(-)	0.02	0.02	0.02

radius is given. It is also possible to generate structures with elongation or stretching in one spatial direction. For instance, the parameter $j \in (0, 1]$ (mentioned in section 3.1) can be used to specify the desired spatial direction. As Figure 4.7 shows, j is defined as the ratio of $\frac{b}{a}$. This has the consequence that the structures are stretched in one direction and thus the mean pore area in the plane orthogonal to the stretched spatial direction is smaller than the mean pore area in the two remaining spatial directions. There is a high probability of pores in real porous structures that stretch in certain directions. For this reason, it is also important to observe the relationship between the stretching degree and the PPI. In this thesis, the structures with pore stretching in one direction as well as two directions are studied.

4.4.1. Voronoi tessellation

The tool *generatemembrane* is based on Voronoi tessellation, so the distribution of Voronoi regions also affects the porous structures. In Figure 4.8 two types of Voronoi tessellations are shown. The uniformity of centroidal Voronoi tessellation is greater than other one [13]. The phenomenon becomes more obvious with larger numbers of Voronoi points.

In this thesis, the most studied structures are based on centroidal Voronoi tessellation. Because of the uniformly distributed Voronoi regions (less particularly large or small regions), when the spheres are placed at the boundary, i.e., during the generation of struts, the Voronoi points

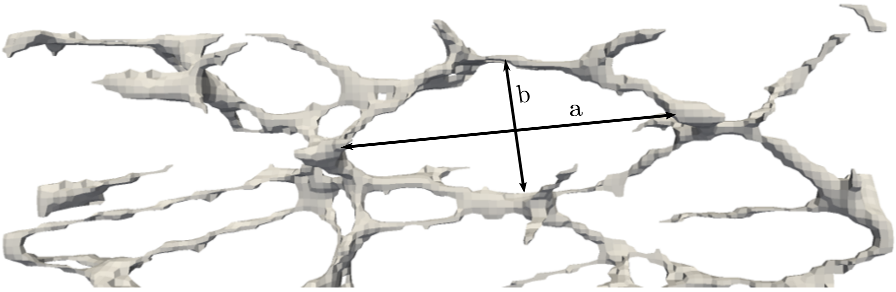


Figure 4.7.: The generated structure with targeted stretching in one spatial direction.

of smaller regions are less frequently covered. When the number of pores in the structure is consistent with the set Voronoi points, it is more advantageous for subsequent analysis, such as finding the relationship between the number of Voronoi points and PPI. In addition, a "large model" (described in section 4.4.3) shows that the centroidal Voronoi tessellations work better for this thesis.

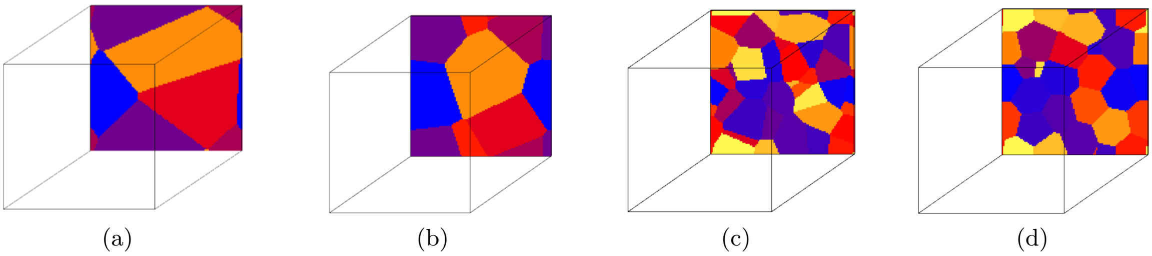


Figure 4.8.: The Voronoi tessellations: (a) a Voronoi tessellation with 10 random points as generators; (b) a centroidal Voronoi tessellation with 10 points as generators, which are also the mass centroids of Voronoi regions; (c) a Voronoi tessellation with 100 random points as generators; (d) a centroidal Voronoi tessellation with 100 points as generators.

4.4.2. Sturcture generation

There are two main types of structures studied in this work. The input parameters to generate the first type of structures are the number of Voronoi points (100, 150, 200 and 250) and porosity (10%-90%). Similar to the first type, the number of Voronoi points is given for the generation of the second type, but instead of the porosity, the strut radius is used and varies from 1 to 5. In the environment PACE3D, structures are composed using voxel cells, so the unit of strut radius is a cell. Figure 4.9 displays a few generated structures and their segmented models.

4.4.3. Validation of computer-generated structures

Validation is also a necessary step before computing PPI. Only a clear and correct structure can ensure that the later computed PPI can be correctly related to the other parameters.

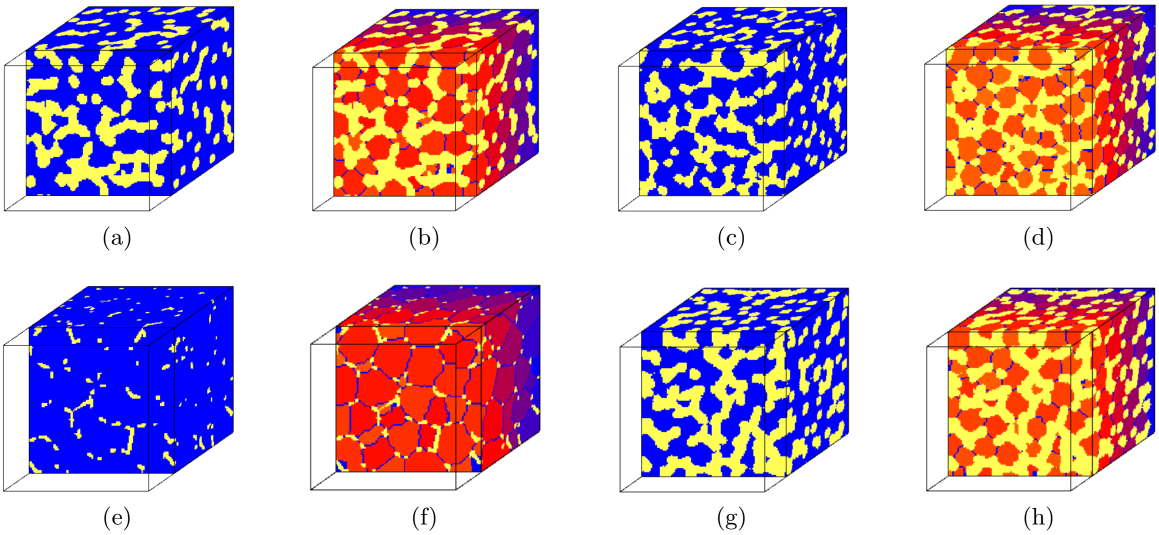


Figure 4.9.: (a), (c), (e) and (g) are the generated structures and (b),(d), (f) and (h) are their segmented models, computed by the marker-based watershed algorithm, respectively, with the input parameters: (a) porosity 60% and 150 Voronoi points; (c) porosity 60% and 250 Voronoi points; (e) strut radius 1 cell and 100 Voronoi points; (g) strut radius 3 cells and 200 Voronoi points. The yellow part represents the solid structures.

If the computed data is not the same as the setting, a decision should be made, which one should be chosen. The following are the approaches to validate:

- porosity: use tool *volume* to check if the porosities of structures are the same as the setting.

- Voronoi points: for this parameter, it needs to be checked whether the number of Voronoi points matches the number of pores.

- strut radius: use the tool *skeletonize* to measure the radius of strut. The radius can be determined from distance fields.

Porosity

Table 4.3 shows the comparison of the set and the computed porosities for different structures, where "set porosity" means the given porosity when using the tool *generatemembrane* and the "computed porosity" is the porosity computed by the tool *volume*. Five structures are generated for each combination of Voronoi points (VP) and set porosity, and the average of porosities is presented. The computed values deviate only slightly from the set values, which is negligible, and the preliminary conclusion is that the obtained porosity of the generated structure is as expected.

Table 4.3.: Comparison of set and computed porosities of different generated structures.

set porosity (-)	Computed porosity (-) for the generated structures with different Voronoi points (VP)			
	250 VP	200 VP	150 VP	100 VP
0.1	0.101	0.102	0.103	0.103
0.2	0.198	0.200	0.200	0.200
0.3	0.297	0.297	0.297	0.299
0.4	0.397	0.396	0.398	0.399
0.5	0.499	0.496	0.497	0.500
0.6	0.597	0.596	0.598	0.601
0.7	0.698	0.700	0.703	0.701
0.8	0.790	0.798	0.798	0.803
0.9	0.905	0.897	0.891	0.888

Voronoi points (the number of pores)

To verify the number of pores, the concept of a "large model" is used. In a large model, the original structure will be extended and copied in three directions. In this way, at the center, the pores on the boundary of the original structure can be interconnected and will not be over-divided when performing the marker-based watershed segmentation. Only the pores in the center, i.e., the domain of the original structure, one twenty-seventh of the large model, will be measured. Figure 4.11 is the validations for the used structures. When the porosity is less than 20% or the strut radius greater than 5% of the domain (5 of 102), the number of pores differs significantly from the number of Voronoi points. The reasons are roughly the same for both situations, as some small pores can be covered when the strut radius is relatively too large or the porosity is too low. In the analysis of the results in chapter 5, this type of structure should be avoided as far as possible to ensure the accuracy of the data.

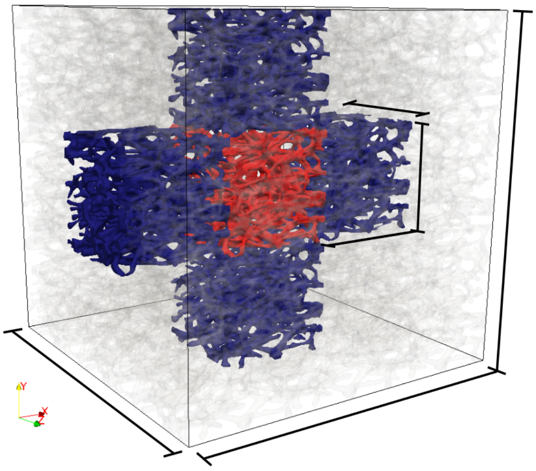


Figure 4.10.: The diagram of a "large model" [23].

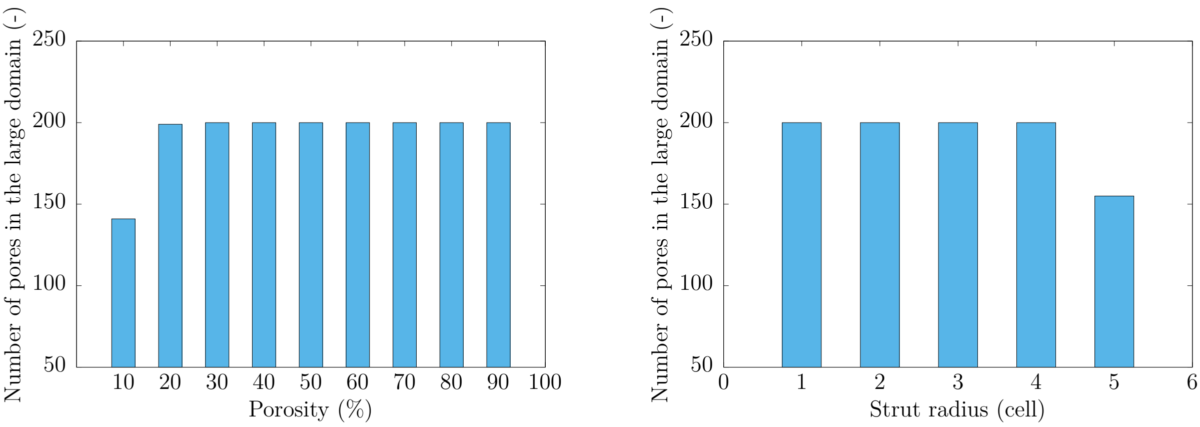


Figure 4.11.: Validation of pore numbers: (a) a validation of porosity-given structures with 200 Voronoi points; (b) a validation of strut-given structures with 200 Voronoi points.

The large model can also be used to compare applicability between the centroidal Voronoi tessellation and non-centroidal Voronoi tessellation. The fit functions in Figure 4.12 show the validation of pore numbers for structures, which are based on two aforementioned Voronoi tessellations. The x-axis is the number of set Voronoi points, y-axis represents the number of pores, which are measured in the center of the large model. The slope of the fit function for structures based on centroidal Voronoi tessellation is close to 1. That means, in such structures, the obtained number of pores is the same as the number of Voronoi points. However, the fit function for geometries based on non-centroidal Voronoi tessellation has a smaller slope and the measured number of pores for structures generated with the same parameters has large fluctuations. In other words, the measured numbers are smaller than the set numbers. With the aid of Figure 4.12, it can be explained better, why the structures with centroidal Voronoi tessellations are studied in this thesis.

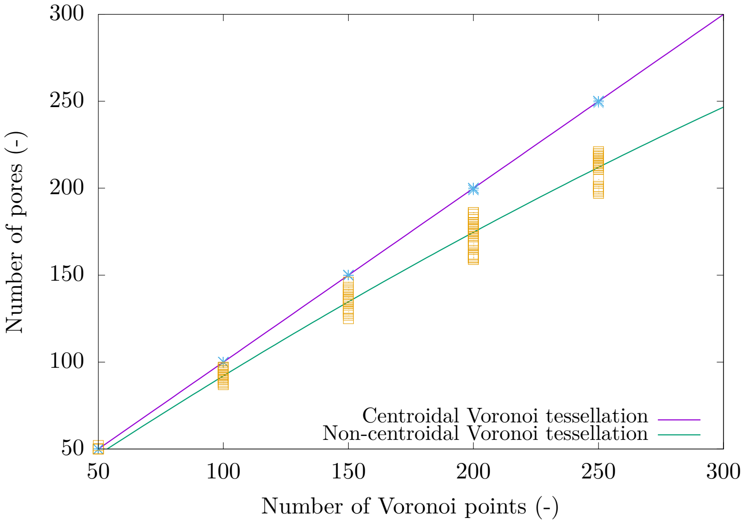


Figure 4.12.: Comparison of pore numbers in the large model between two different Voronoi tessellations.

Strut radius

Table 4.4 displays the comparison of the set and computed strut radii for different structures, where set means the given radius when using the tool *generatemembrane* and the computed radius is the radius computed by the tool *skeletonize*. Because of the differences in algorithms, the computed strut radius differs from the corresponding set radius and this difference becomes even larger when the structure has more pores. The main focus of this work is to observe the trend in PPI. A small amount of deviation will not lead to a big change in trend, so the set strut radius is chosen as the object of observation.

Table 4.4.: Comparison of set and computed strut radius (cell) of different generated structures.

set strut radius (cell)	Computed strut radius (cell) for the generated structures with different Voronoi points (VP)			
	250 VP	200 VP	150 VP	100 VP
1	1.204	1.200	1.194	1.184
2	2.360	2.357	2.346	2.334
3	3.114	3.161	3.190	3.263
4	3.588	3.644	3.785	3.957

4.5. Real structures

The relationship between PPI and the other parameters can already be obtained through the more complex computer-generated structures. The next stage of this work is to put this tool into practice: reconstruct the real structure and compare the results of the computer simulation with the data given by the manufacturer. The samples used in this work are open-cell aluminium foams ($AlSi_7Mg$) with pore densities of 10 PPI and 30 PPI and the porosities of the two samples are 88.441% and 90.999%[14]. The porosity is used as a criterion for processing images to determine whether the reconstructed structures are close to the true value.

4.5.1. Reconstruction

Since the steps for processing the images and reconstructing the structures are the same for each model, here a structure with 10 PPI is taken as an example, the procedure is as follows. Figure 4.13 shows the cross-section images of the structure in various stages:

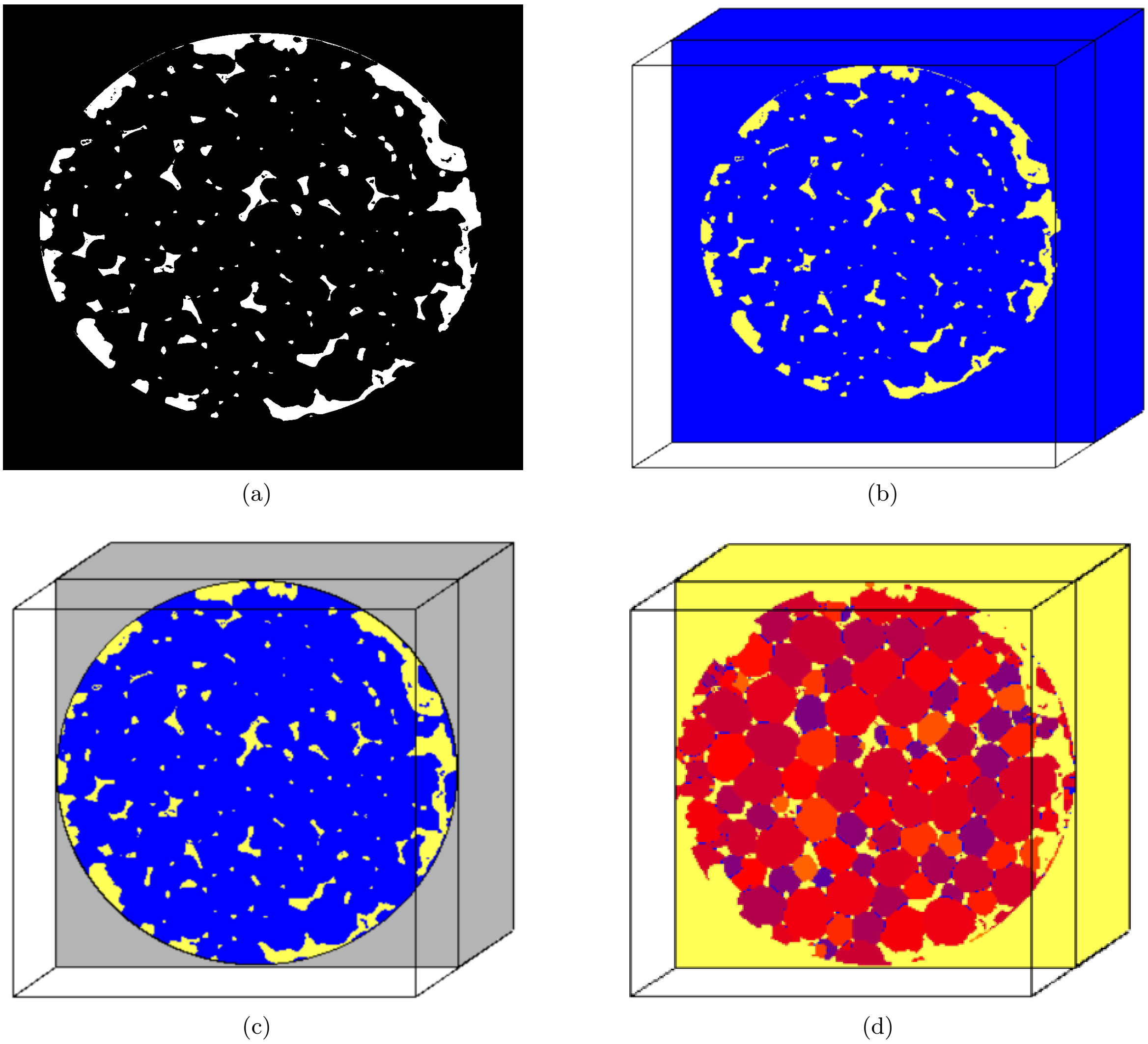
1. As mentioned in section 3.4, the CT images of specimen are processed using Fiji ImageJ. The entire CT image stack contains low-quality images, in which the pores are poorly defined, that did not meet the requirements for reconstruction in this work. Only the clear part of the images stack is used.

2. The image stack is converted into a binary image stack and an appropriate threshold is selected to ensure that the porosity of the entire image stack is close to the experimentally given porosity.

3. Using the tool *slices2scalardata*, a set of binary images of cross-sections of a specimen is combined and transformed into a 3D model, the pixels in the images become voxels in the 3D model.

4. Validation of porosity in the 3D model is necessary. The reconstructed 3D model is cut to a suitable size and the surrounding part that not the part of the foam is transformed into barriers (no influence on porosity).

5. The pore network model of validated structures can be extracted by tool *extractnetwork*, and then the PPI can be computed.


(a)

(b)

(c)

(d)

Figure 4.13.: The reconstruction process of real structures with 10 PPI: (a) a processed binary image of the cross-section of real structure; (b) the reconstructed 3D model; (c) the structure, whose surroundings are transformed into barriers; (d) the segmented model of the reconstructed structure.

4.5.2. Validation of porosity

The porosity of the 3D model needs to be verified before computing the PPI. When the porosity of the images stack is adjusted close to the experimental data, the porosity of the 3D model is larger than the experimental data. Therefore, by image processing, the porosity of the image stack is deliberately reduced to lower than the experimental data and then the porosity of the 3D model can be obtained, which is closer to the experimental value. As the 3D models are used directly for computing PPI in this work, it is important to keep the porosity of the 3D model closer to the experimental value. For the structures with 10 PPI, a reconstructed 3D model with 88.39% porosity is used. For the structure with 30 PPI, a reconstructed 3D model with 90.83% porosity is used, the cross-sections of the structure are displayed in Figure 4.14. Table 4.5 shows the results of validation of porosity.

For the reconstructed structures, firstly, the original algorithm is used when computing PPI in which the length of the measuring line is constant and equivalent to the length of the domain. From the cross-section images of the pore network model, it can be seen that the periphery (yellow part) of the foam cannot be completely removed. If the length of the domain acts as the length of the measuring line, the result may be affected. Similar to reducing the impact of periodic boundaries, a second approach is therefore proposed, with all measuring lines starting at the second pore and ending at the penultimate pore. The results of the two approaches are compared in chapter 5.

Table 4.5.: Validation of porosity.

	experimental porosity (%)	porosity of 3D model (%)
10 PPI	88.44 [14]	88.39
30 PPI	90.99 [14]	90.83

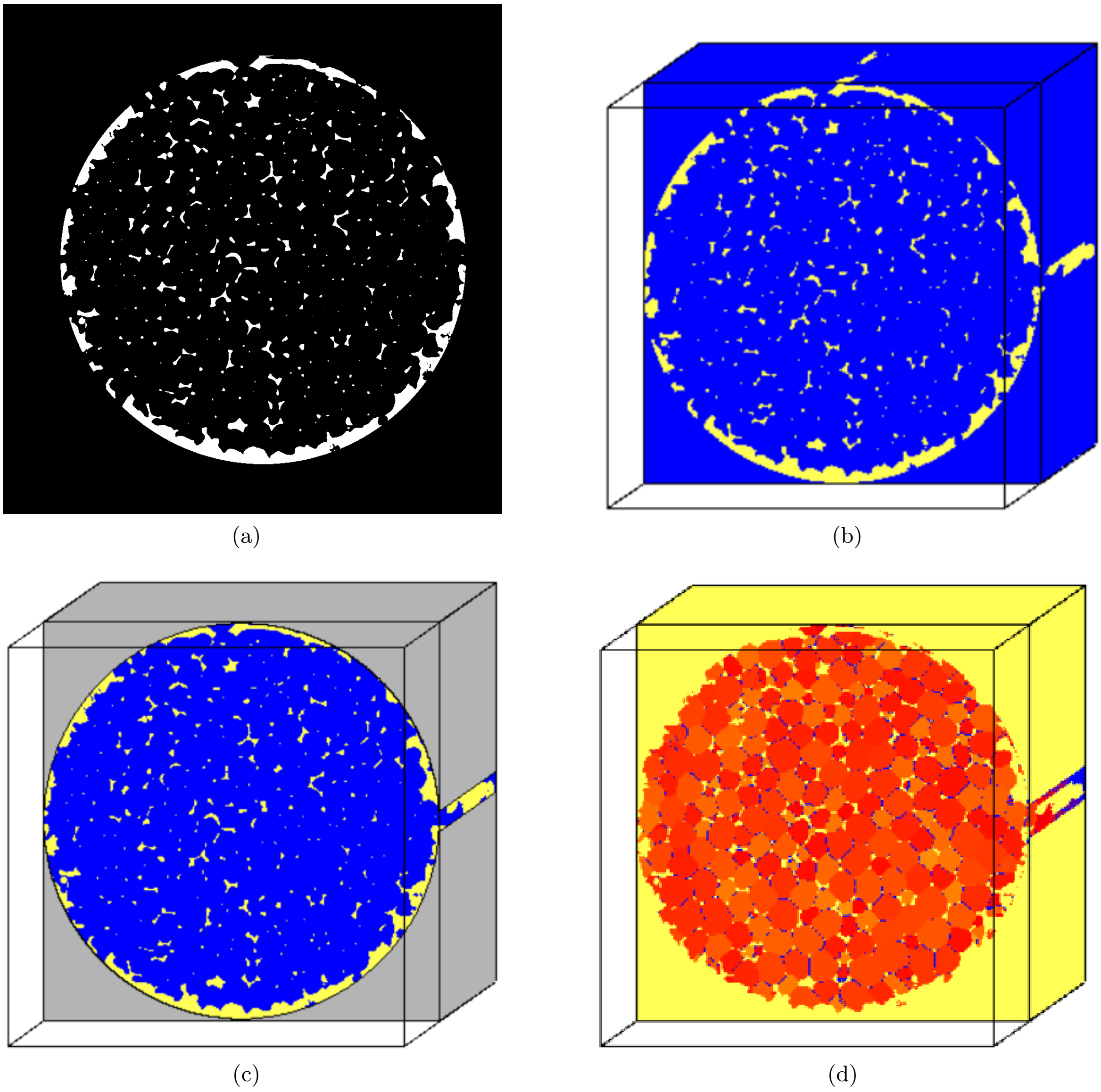


Figure 4.14.: The Reconstruction of structure with 30 PPI: (a) a processed binary image of the cross-section of real structure; (b) the reconstructed 3D model; (c) the structure, whose surroundings are transformed into barriers; (d) the segmented model of the reconstructed structure.

5. Results

This chapter shows the results of the PPI counting algorithm in computer-generated structures. The relationships between other parameters and PPI are discussed. The PPI values of the real structures are also computed by using reconstructed structures.

5.1. PPI for complex computer-generated structures

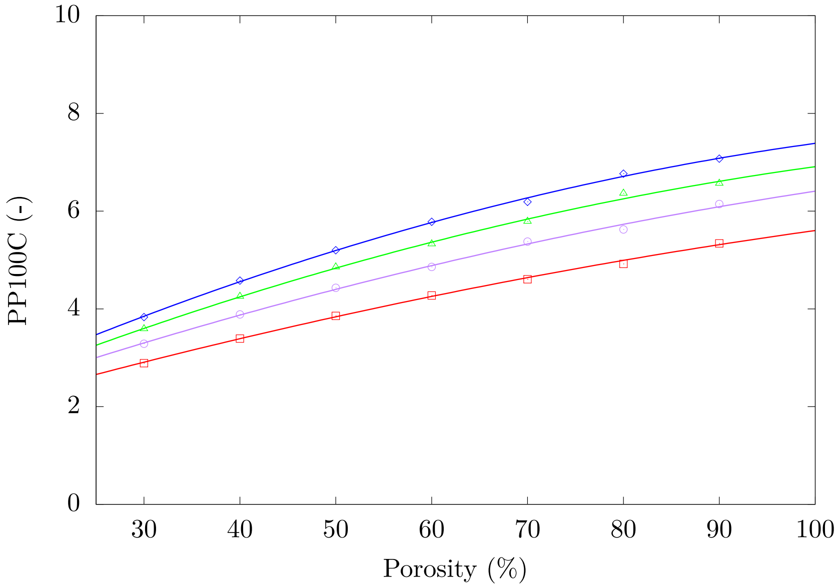
5.1.1. Porosity-given structures

The first modified algorithm

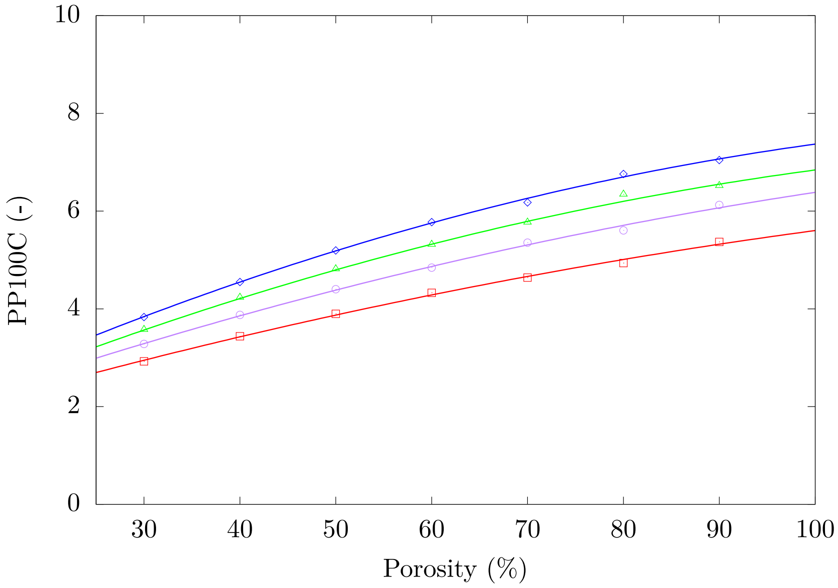
Figure 5.1 shows the number of pores per 100 cells computed with the first modified algorithm (see section 4.3), in the x-direction, y-direction, z-direction and the average PP100C for the entire volume, respectively. As the graphs show, the variation of PP100C, i.e., PPI is in good agreement with expectation and the results in the three directions are not significantly different. The structures with the same number of Voronoi points (pores) but different porosities have different PP100C. The computed PP100C increases with a larger porosity. When the structures have the same porosity, the more Voronoi points they contain, the greater the computed PP100C is. In structures with a low total number of Voronoi points, the fitting function between PP100C and porosity is close to a primary function, namely linear, which means the rate of increase of PP100C with porosity is almost constant. When the total number of Voronoi points in the structures is greater, the fitting function between porosity and PP100C is more in line with a quadratic function, the growth rate of PP100C is at low porosities faster than at high porosities.

In addition, in the structures generated with the same conditions, the first modified algorithm runs quickly, the error of the PP100C computed with the first modified algorithm is very small and the amplitude of error does not change with different porosities or number of Voronoi points. The results are also similar in all directions (see the average of PP100C in Table 5.1). For the structures with 250 pores, the average PP100C varies from 3.83 to 7.07; for the structures with 200 pores, that varies from 3.59 to 6.55; for structures with 150 pores, the average PP100C varies from 3.28 to 6.16; for structures with 100 pores average PP100C varies from 2.92 to 5.39. As the number of Voronoi points in the structures increases, the magnitude of the variation in PP100C also increases.

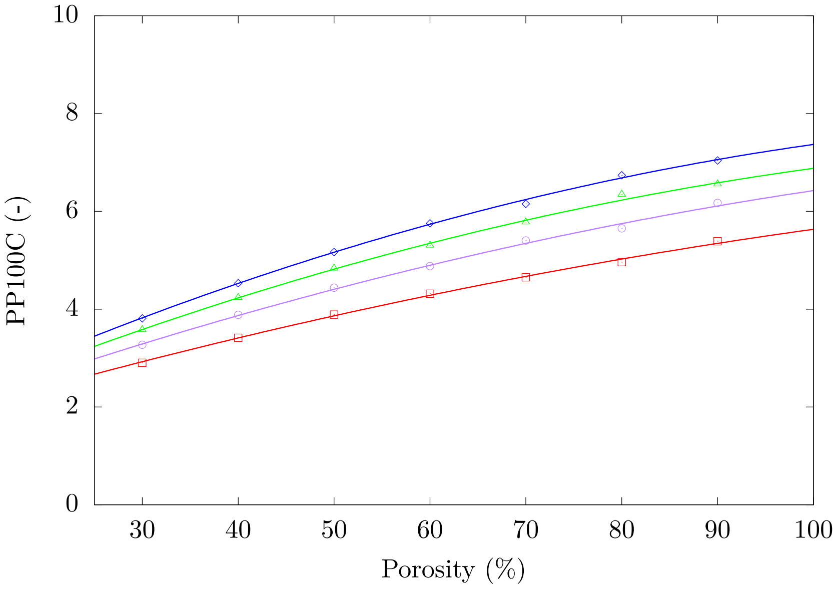
Number of Voronoi points	Computed PP100C	Fit function
100	□	—
150	○	—
200	△	—
250	◇	—



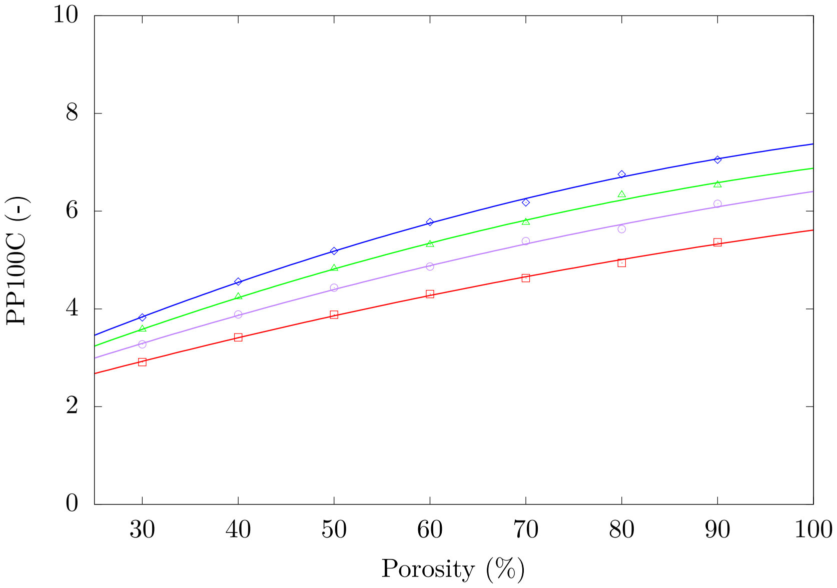
(a)



(b)



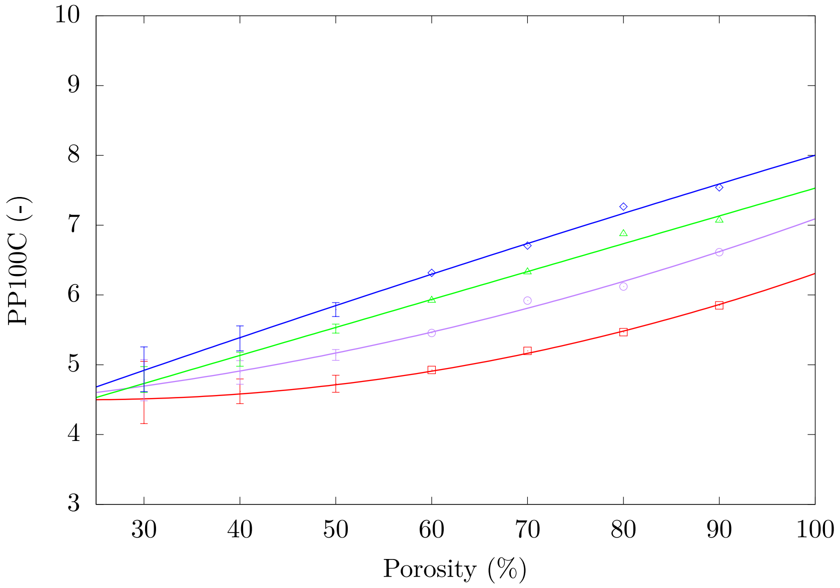
(c)



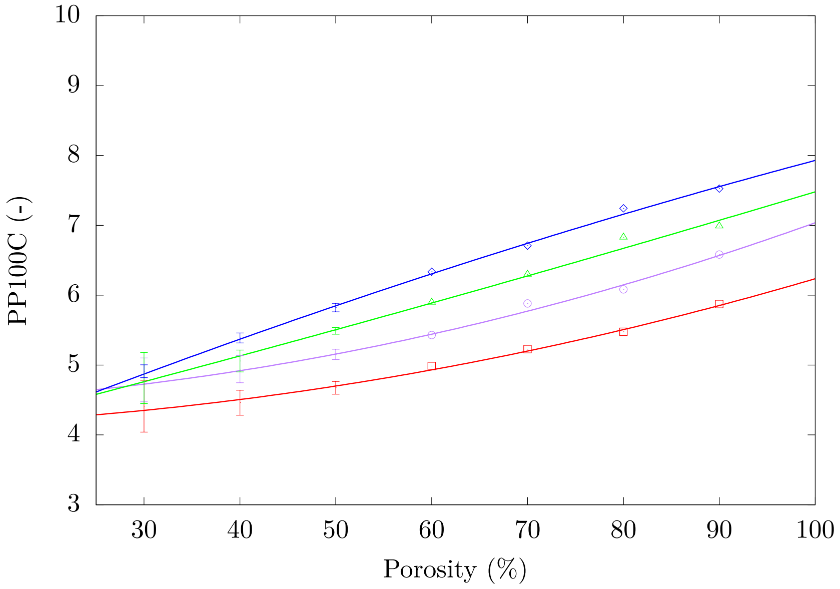
(d)

Figure 5.1.: The number of pores per 100 cells (PP100C) for structures with different porosities computed using the first modified algorithm: (a) PP100C in the x-direction; (b) PP100C in the y-direction; (c) PP100C in the z-direction; (d) average PP100C for the entire volume. Each point represents data from five structures with different seeds.

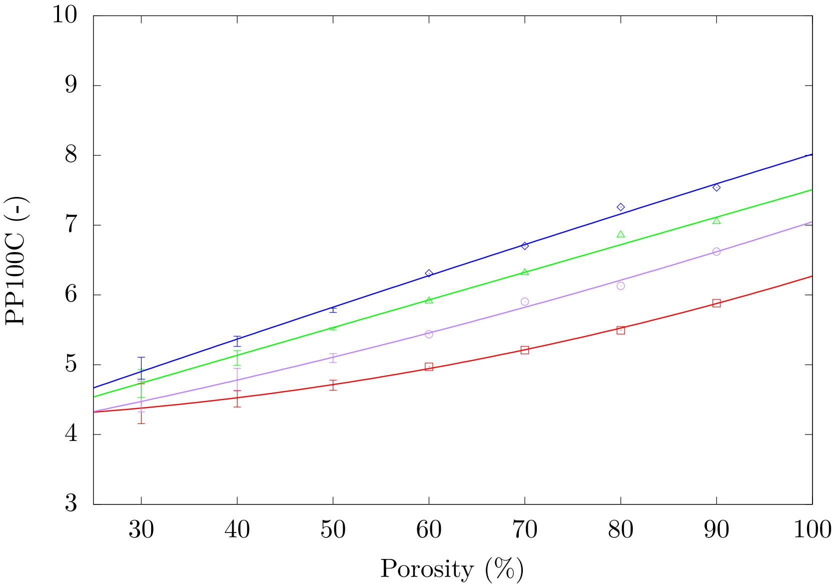
Number of Voronoi points	Computed PP100C	Fit function	Error bars
100	□	—	├─•─┤
150	○	—	├─•─┤
200	△	—	├─•─┤
250	◇	—	├─•─┤



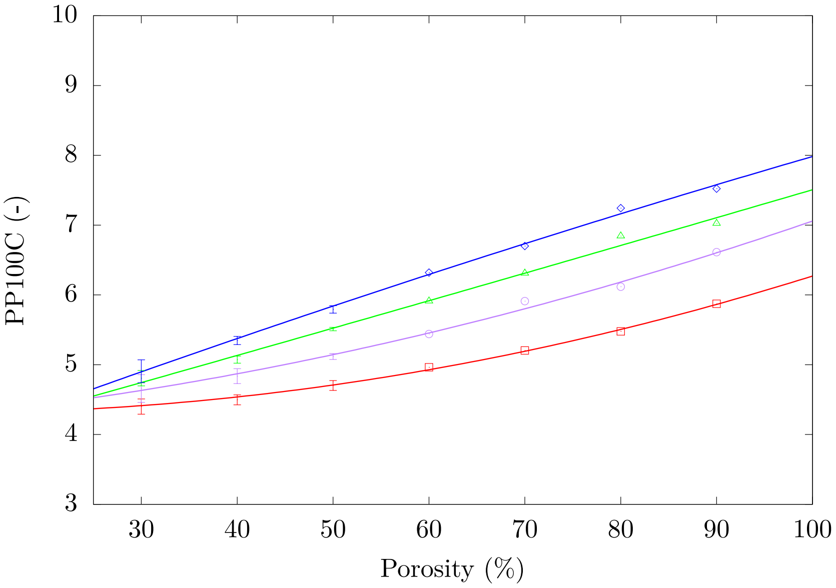
(a)



(b)



(c)



(d)

Figure 5.2.: The number of pores per 100 cells (PP100C) for structures with different porosities computed using the second modified algorithm: (a) PP100C in the x-direction; (b) PP100C in the y-direction; (c) PP100C in the z-direction; (d) average PP100C for the entire volume. Each point represents data from five structures with different seeds.

Table 5.1.: The average PP100C computed by the first algorithm for the porosity-given structures.

porosity (%)	Computed PP100C (-) by structures with different Voronoi points (VP)			
	250 VP	200 VP	150 VP	100 VP
30	3.83	3.59	3.28	2.92
40	4.57	4.25	3,89	3.43
50	5.20	4.85	4.44	3.89
60	5.78	5.32	4.87	4.32
70	6.20	5.78	5.40	4.65
80	6.77	6.35	5.64	4.96
90	7.07	6.55	6.16	5.39

The second modified algorithm

Figure 5.2 shows the PP100C computed with the second modified algorithm (see section 4.3), also in the x-direction, y-direction, z-direction and the average PP100C for the entire volume, respectively. There are no significant differences in the results in the three directions. The PP100C computed by the second modified algorithm also increases with porosity and the number of Voronoi points, but the specific values and growth rate are different from those computed by the first modified algorithm. In the structures with a smaller number of Voronoi points, at first, the PP100C does not vary much with porosity, until porosity is greater than 60%, then the tendency for PP100C to increase is more pronounced. In structures with a higher number of Voronoi points, the rate of increase for PP100C is more stable. In structures with porosities less than 50%, the results of PP100C have large errors. Presumably, the reason for this is that the second modified algorithm is sensitive to the distribution of pores. Because the length of the measuring line is not constant, the PP100C is large, when multiple pores are distributed on a short measuring line whereas the PP100C is small on a long measuring line with fewer pores. Therefore, the second modified algorithm is not suitable for structures with small numbers of Voronoi points and low porosities, it may cause great errors by measuring. Conversely, when both the number of Voronoi points and the porosity in the structures are high, the results are more accurate and stable. Table 5.2 displays the average PP100C computed by the second modified algorithm. For the structures with 250 pores, the average PP100C varies from 4.93 to 7.53; for the structures with 200 pores, that varies from 4.80 to 7.03; for structures with 150 pores, the average PP100C varies from 4.67 to 6.61; for structures with 100 pores, the average PP100C varies from 4.43 to 5.88.

Comparison of the two modified algorithms

Firstly, both modified algorithms are considered to be applicable to the periodic structures and can partly exclude some influence of periodic boundaries on the results. But both algorithms have their advantages and drawbacks. Figure 5.3 shows the PP100C computed with the two modified algorithms, the Voronoi points are 150 and 250, respectively. As can be seen, the results computed by the second modified algorithm are generally larger than those computed

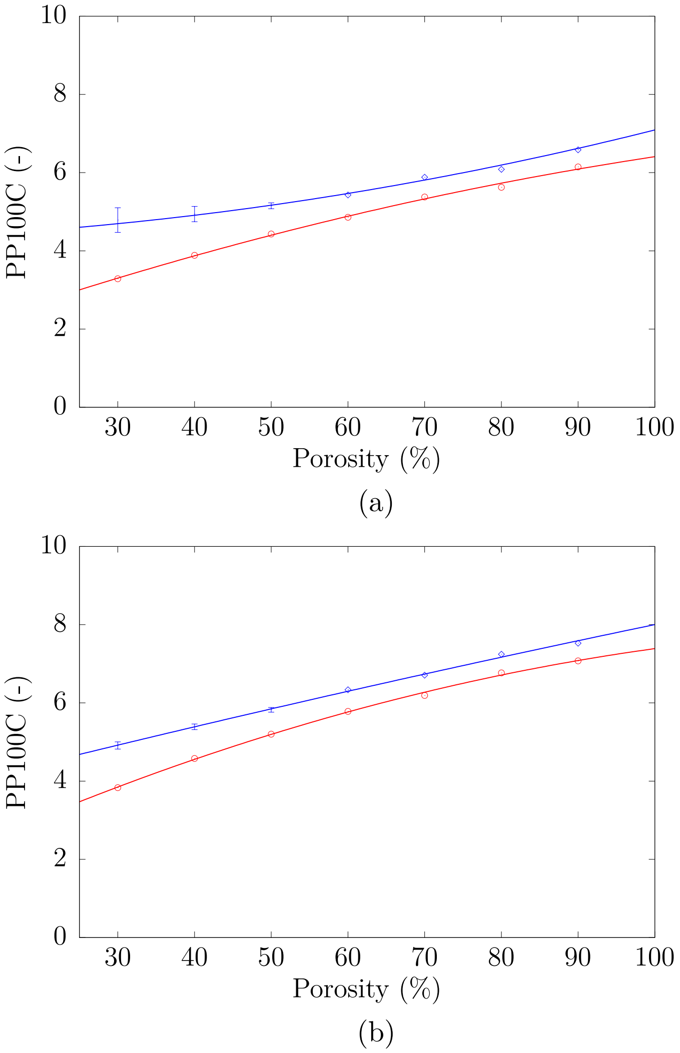
Table 5.2.: The average PP100C computed by the second modified algorithm for porosity-given structures.

porosity (%)	Computed PP100C (-) by structures with different Voronoi points (VP)			
	250 VP	200 VP	150 VP	100 VP
30	4.93	4.80	4.67	4.43
40	5.35	5.07	4.83	4.51
50	5.80	5.51	5.12	4.70
60	6.32	5.90	5.45	4.96
70	6.70	6.31	5.91	5.20
80	7.25	6.84	6.12	5.47
90	7.53	7.03	6.61	5.88

by the first modified algorithm. The reason is, it is too simple in the first modified algorithm to subtract 1 directly from the number of pores and not change the length of measuring lines, which can lead to smaller computed PP100C.

The difference varies with porosities; for structures with the same number of Voronoi points but lower porosities, the difference between computed PP100C is larger. For instance, the difference between PP100C for structures with 30% porosity is around 2, but the difference for structures with 90% porosity is around 1.3. In addition, for structures with a large number of Voronoi points, the difference caused by the two algorithms is smaller than for structures with fewer Voronoi points. Table 5.3 shows the comparison of the average PP100C by two modified algorithms for structures with different Voronoi points. The porosity of all used structures is 90%, the domain of the first 4 types of structures, i.e., structures with 100, 150, 200 and 250 Voronoi points, is 100 × 100 × 100 cells. The domain of the other structures is 200 × 200 × 200 cells. When a large number of Voronoi points are placed in a larger domain, the accuracy of the generated structures is better guaranteed. Arranging 250 Voronoi points in a structure with a domain of 100 × 100 × 100 cells is theoretically the same as arranging 2000 Voronoi points in a structure with a domain of 200 × 200 × 200 cells. However, for the latter (more Voronoi points in a larger domain), the relative difference between the two modified algorithms is smaller, which indicates that both modified algorithms are more accurate in the structures with a large number of Voronoi points. The relative difference between the two algorithms decreases with the increasing number of Voronoi points in the structure. When 2000 or more Voronoi points are placed in the entire structure, the relative difference can be less than 3%. Additionally, the results obtained by the first modified algorithm can give a clearer picture of the relationship and trend between PP100C and porosity. There are no errors between the results for structures generated with the same conditions. As mentioned before, the disadvantage of the second modified algorithm is obvious, the results are susceptible to the distribution of pores at porosities less than 50%, the errors of results are large. Conversely, the advantage of the second modified algorithm is that more accurate values can be obtained in structures with large porosity and a high number of Voronoi points, this condition usually can be met by porous materials used in real life.

Algorithm	Computed PP100C	Fit function	Error bars
The first	○	—	none
The second	◇	—	├•┤



(a)

(b)

Figure 5.3.: The number of pores per 100 cells (PP100C) in x-direction for structures with different porosities computed using two different modified algorithms: (a) the number of pores is 150; (b) the number of pores is 250.

Table 5.3.: The average computed PP100C by the two modified algorithms.

The modified algorithm	Computed PP100C by structures with different Voronoi points (VP)							
	domain 100 × 100 × 100				domain 200 × 200 × 200			
	100 VP	150 VP	200 VP	250 VP	2000 VP	3000 VP	4000 VP	5000 VP
The first modification	5.39	6.16	6.55	7.07	7.18	8.09	8.92	9.42
The second modification	5.88	6.61	7.03	7.53	7.4	8.31	9.15	9.66
The relative difference (%)	8.33	6.81	6.83	6.11	2.98	2.65	2.51	2.48

5.1.2. Strut radius-given structures

The first modified algorithm

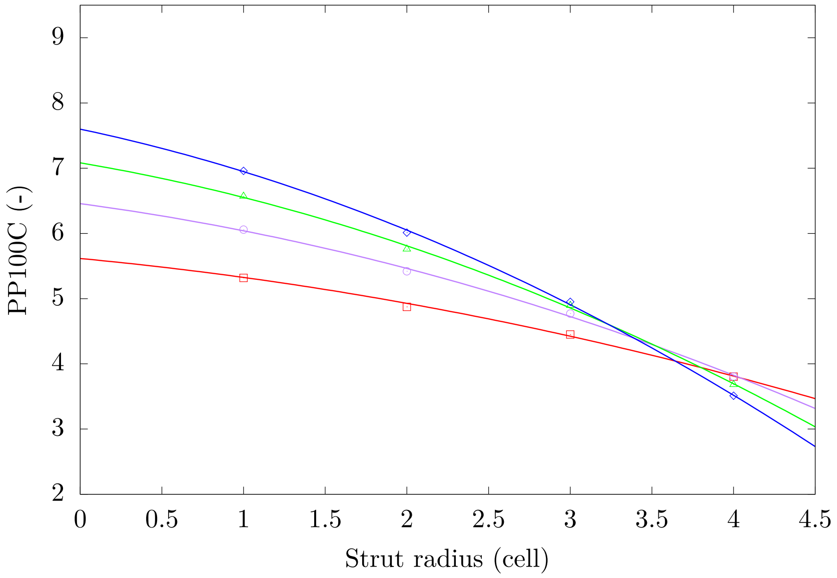
Figure 5.4 shows the results of PP100C computed by the first modified algorithm (see section 4.3) and the relationship between PP100C and strut radius, in the x-direction, y-direction, z-direction and the entire volume, respectively. The graphs show that when a structure contains more Voronoi points, this structure is more sensitive to the changes in strut radius and the PP100C value decreases significantly as the strut radius increases. Conversely, structures with a smaller number of Voronoi points have a smaller variation of PP100C. As for the porosity-given structures, the error of the results computed by the first algorithm is small, the results do not differ much in the three spatial directions and in Table 5.4, the average values of PP100C are shown. For the structures with 250 pores, the average PP100C varies from 6.97 to 3.50; for the structures with 200 pores, that varies from 6.55 to 3.67; for structures with 150 pores, the average PP100C varies from 6.07 to 3.81; for structures with 100 pores, the average PP100C varies from 5.37 to 3.83. It is important to be noted, as shown in graphs, there is a critical range around the strut radius of 3.5. When the strut radius of the structure is less than the critical range, more Voronoi points (pores) in the structure result in greater values of PP100C. When the strut radius is greater than the critical range, the previous rule does not hold again.

Table 5.4.: The average PP100C computed by the first algorithm for strut radius-given structures.

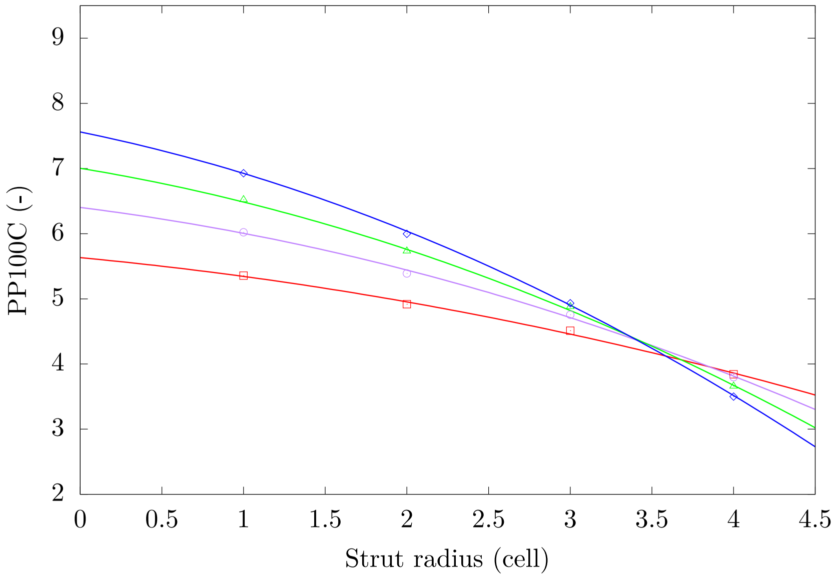
Strut radius (cell)	Computed PP100C (-) by structures with different Voronoi points (VP)			
	250 VP	200 VP	150 VP	100 VP
1	6.97	6.55	6.07	5.37
2	6.00	5.75	5.42	4.92
3	4.95	4.90	4.78	4.50
4	3.50	3.67	3.81	3.83

To analyze the reason for the overall trend, the strut radius can correlate with porosity. Table 5.5 shows the average porosities of strut radius-given structures. As shown, in a structure with a large number of Voronoi points (pores), the porosity changes from 88.25% to 23.52%, while the strut radius changes from 1 to 4. In the structures with fewer Voronoi points, the porosity changes only from 93.3% to 46.30%. This can explain why structures with a higher number of Voronoi points are more sensitive to the changes in strut radius because the changes can lead to a greater difference in porosity. When the strut radius is 1, there is not much distinction in the porosity of structures with different numbers of pores (88.25% ~93.3%), now the principle should be followed: the higher the number of Voronoi points, the greater PP100C value. When the strut radius is 4, the gap of porosity in different structures becomes wider (23.52% ~46.3%), now the number of Voronoi points plays no longer a decisive role. That can explain why there is a critical range.

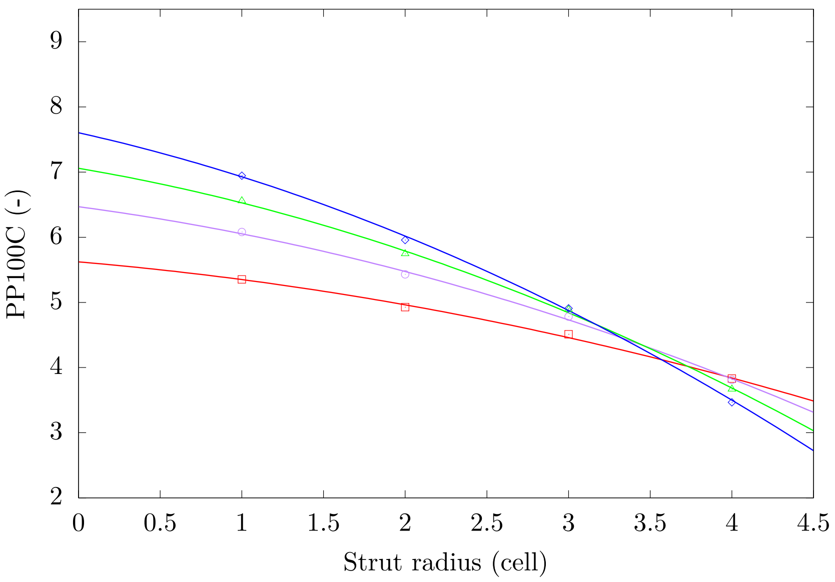
Number of Voronoi points	Computed PP100C	Fit function
100	□	—
150	○	—
200	△	—
250	◇	—



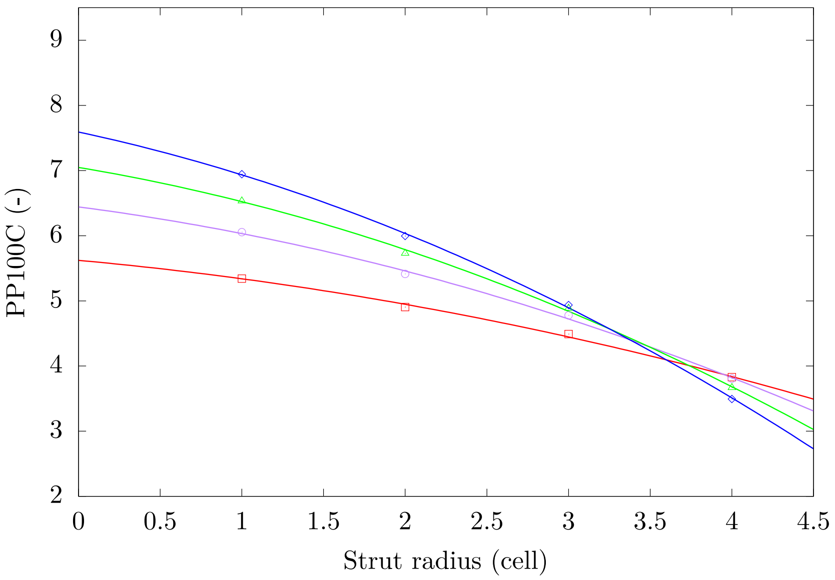
(a)



(b)



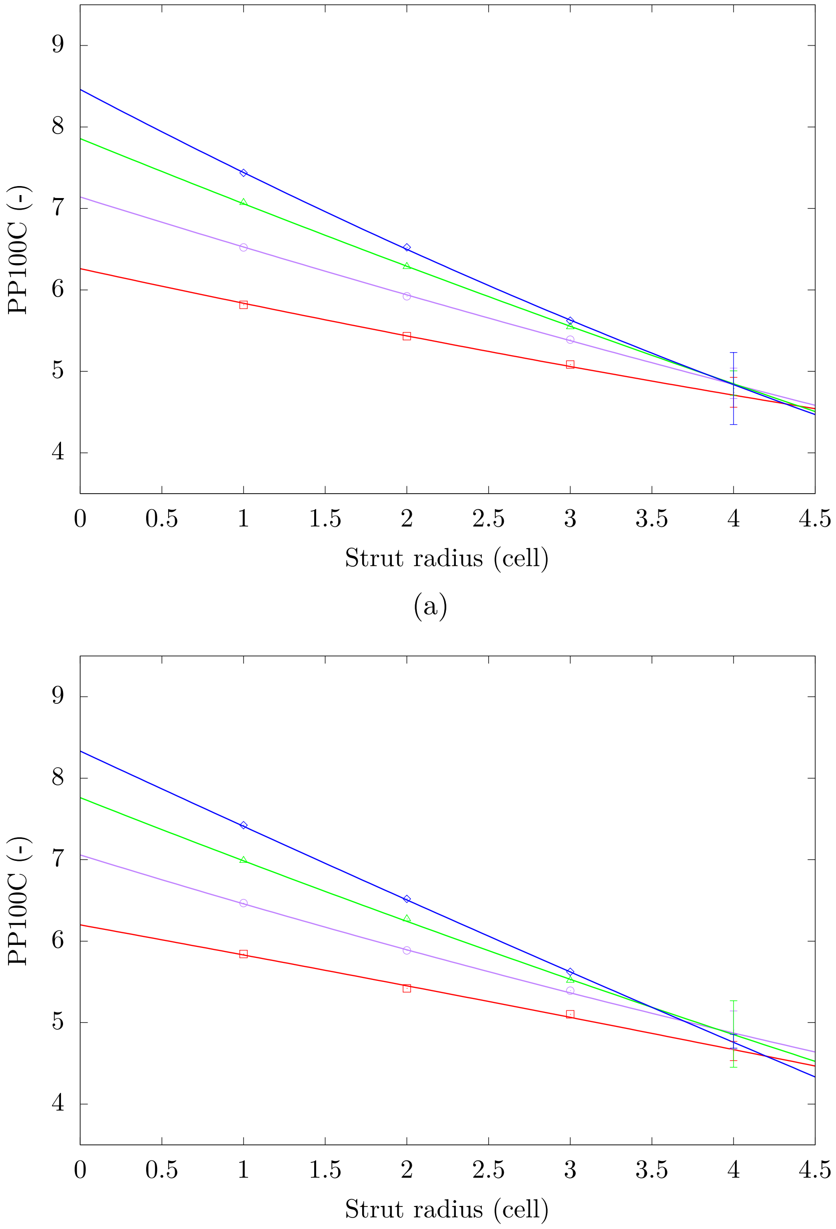
(c)



(d)

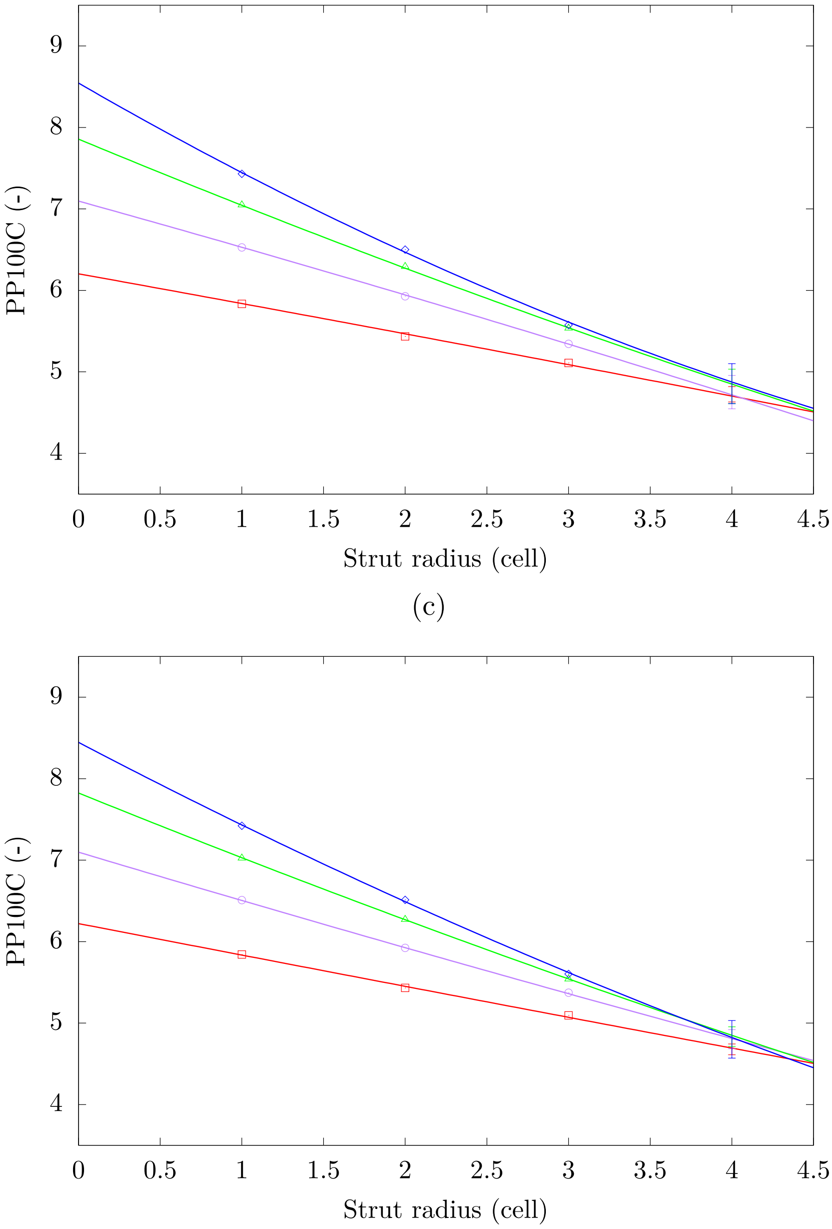
Figure 5.4.: The number of pores per 100 cells (PP100C) for structures with different strut radii computed using the first modified algorithm: (a) PP100C in the x-direction; (b) PP100C in the y-direction; (c) PP100C in the z-direction; (d) average PP100C for the entire volume. Each point represents data from five structures with different seeds.

Number of Voronoi points	Computed PP100C	Fit function	Error bars
100	□	—	⊢•⊣
150	○	—	⊢•⊣
200	△	—	⊢•⊣
250	◇	—	⊢•⊣



(a)

(b)



(c)

(d)

Figure 5.5.: The number of pores per 100 cells (PP100C) for structures with different strut radii computed using the second improved algorithm: (a) PP100C in the x-direction; (b) PP100C in the y-direction; (c) PP100C in the z-direction; (d) average PP100C for the entire volume. Each point represents data from five structures with different seeds.

Table 5.5.: The average porosity of strut radius-given structures.

Strut radius (cell)	Computed porosity (%) by structures with different pore numbers			
	250 VP	200 VP	150 VP	100 VP
1	88.25	89.72	91.36	93.30
2	65.60	69.40	73.78	79.19
3	43.66	48.93	55.30	63.64
4	23.52	28.99	36.12	46.30

The second modified algorithm

Figure 5.5 shows the results computed by the second modified algorithm (see section 4.3). The specific trend is similar to that obtained by the first modified algorithm, but at large strut radii such as 4 cells, the measuring error of the second algorithm increases significantly. This is consistent with the previous conclusion that the second modified algorithm is more suitable for structures with high porosity and a larger number of Voronoi points. Table 5.6 shows the average PP100C computed by the second modified algorithm, which is generally bigger than the result of the first modified algorithm. For the structures with 250 pores, the average PP100C varies from 7,42 to 4.83; for the structures with 200 pores, that varies from 7.03 to 4.85; for structures with 150 pores, the average PP100C varies from 6.51 to 4.81; for structures with 100 pores, the average PP100C varies from 5.84 to 4.69. The previous view holds that the results of the first modified algorithm are suitable for observation trends and the results of the second modified algorithm are suitable for measuring specific values.

Table 5.6.: The average PP100C computed by the second algorithm for strut radius-given structures.

Strut radius (cell)	Computed PP100C (-) by structures with different Voronoi points (VP)			
	250 VP	200 VP	150 VP	100 VP
1	7.42	7.03	6.51	5.84
2	6.51	6.27	5.92	5.43
3	5.60	5.53	5.37	5.10
4	4.83	4.85	4.81	4.69

5.1.3. Structures with stretching in one direction

The distribution and stretching direction of pores affect the mechanical properties of porous materials and the flow of liquids in a porous medium. Therefore, studying the relationship between PPI and the stretching of pores is also beneficial to the characterization of porous structures by PPI. As mentioned in section 5.1.1, when the structure has high porosity and more Voronoi points, the PP100C computed by the two modified algorithms are more accurate and have minor relative differences. As a result, when studying the effect of stretching of pores on PPI, structures containing 4000 Voronoi points with 90% porosity are used. The domain of structures is 200 × 200 × 200 cells. The results are computed by the second modified algorithm. Figure 5.6 shows the PP100C in three spatial directions for structures with stretching in the x-direction.

Direction	Computed PP100C	Fit function
x-direction	□	—
y-direction	○	—
z-direction	△	—

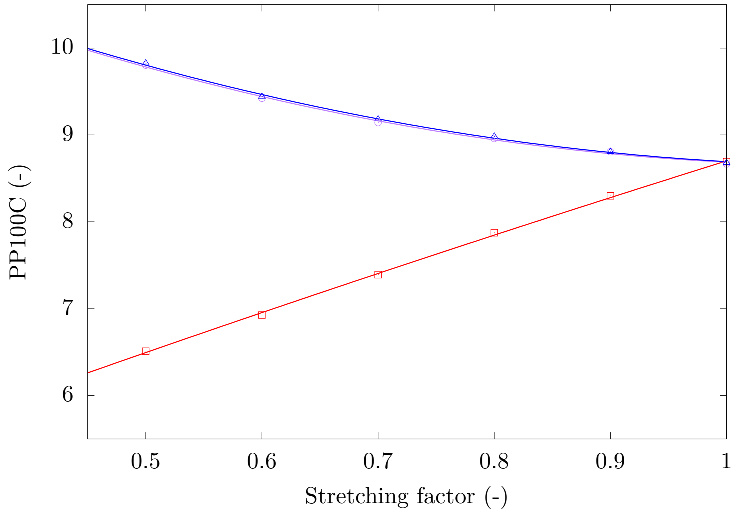
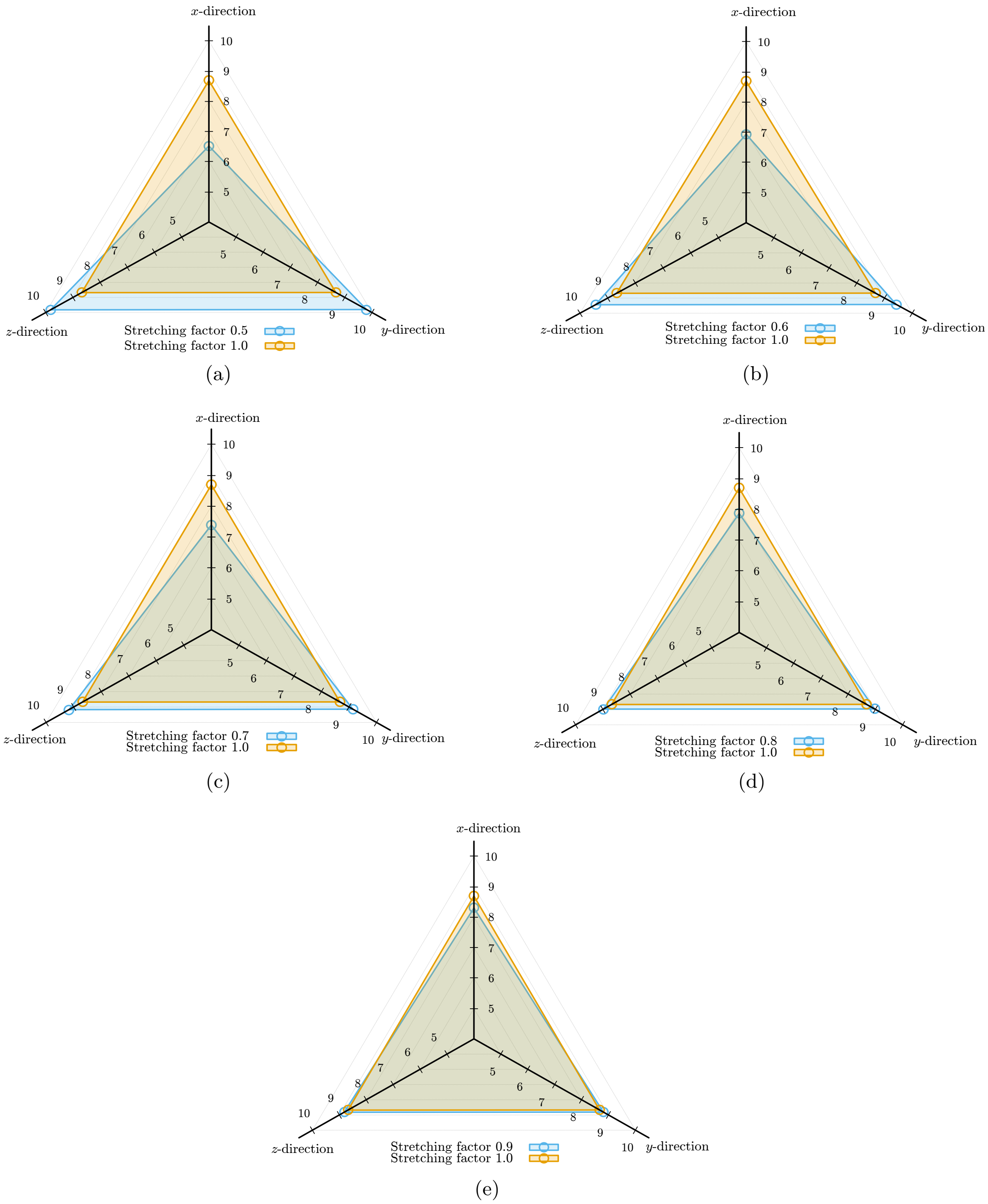


Figure 5.6.: The number of pores per 100 cells (PP100C) for structures with stretching in the x-direction. Each point represents data from five structures with different seeds.

The stretching factor 0.5 means that the radius of pores in the x-direction is close to twice as long as in the other two spatial directions; the stretching factor 1 means that the pores have equal radii in all three spatial directions (no stretching). To keep the porosities constant and the domain of structures unchanged, if the radius of the pores is stretched in one direction, a slight adjustment of the pore radius in the other two directions is necessary.

With a small stretching factor, the PP100C in the stretching direction is much smaller than the PP100C in the other two spatial directions. Since there is no stretching, the PP100C in the other two directions is basically the same. As the stretching factor increases, the PP100C in the stretching direction gradually increases, and on the contrary, the PP100C in the other two directions slightly decreases. When the stretching factor is 1, the PP100C in all directions reaches the same value, the PP100C in all directions is around 8.67. When the stretching factor is 0.5, the PP100C in the x-direction is around 6.52 and in the other two directions is around 9.8. The radar charts in Figure 5.7 show the PP100C of the structures stretched in the x-direction with different factors. In (a) the stretching factor in the x-direction is 0.5, which is a large degree of stretching. It is in clear view that the stretched structures (blue) have a much smaller PP100C in the x-direction than the unstretched structures (sandy brown). In the other two directions, the PP100C of the stretched structures (blue) is slightly larger than that of the unstretched structures (sandy brown). The situation is significantly different in (e), where the stretching factor in the x-direction is 0.9, which is a small degree of stretching.

The PP100C of the stretched structures (blue) and the unstretched structures (sandy brown) are very close to each other, but the PP100C still maintains the previous variation pattern: smaller in the x-direction and larger in the other directions.



(a)

(b)

(c)

(d)

(e)

Figure 5.7.: The PP100C of structures stretched in x-direction with different stretching factors: (a) factor is 0.5; (b) factor is 0.6; (c) factor is 0.7; (d) factor is 0.8; (e) factor is 0.9.

5.1.4. Structures with stretching in two directions

The used structures are stretched in both x and y-directions with stretching factors ranging from 0.5 to 1.0. Figure 5.9 shows the variation of the PP100C with the stretching factor in the y-direction, wherein (a)-(f) the stretching factors in the x-direction are 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, respectively. When the stretching factor in the x and y-directions are identical, the PP100C computed in these two directions are almost the same. It can be observed, that the fitting curves of the PP100C in x and y-directions intersect around the points, where the stretching factors are identical in both directions. When the stretching factors in the x and z-directions are constant, the PP100C computed in the y-direction grows as the stretching factor increases, while the PP100C in the other two directions progressively decreases. In (a)-(e), the PP100C in the x-direction is smaller than that in the z-direction because the pore radius in the x-direction is always stretched (stretching factor varies from 0.5 to 0.9). In (f), the stretching factor in the x-direction is 1.0 (no stretching) and the computed PP100C in both directions becomes the same. It can be preliminarily concluded that in porous structures, the greater the stretching degree in a certain direction (in this thesis means a smaller stretching factor), the smaller PP100C is computed in that direction. Figure 5.8 shows the PP100C of the structures stretched in two directions with stretching factors 0.5 and 0.7.

Under the influence of the stretching factor, the PP100C computed in the stretching direction becomes smaller. In the stretching directions, the PP100C of the structure with a stretching factor of 0.5 is smaller than that of the structure with a factor of 0.7. Additionally, the PP100C in the z-direction is also indirectly influenced by the stretching, the PP100C is larger when the stretching factor is 0.5, while the PP100C is smaller when there is no stretching in the structures. The higher the stretching degree of the pore radius in certain directions, the larger the PP100C computed in the remaining direction. However, for the structures with the same size, porosity and the number of Voronoi points, the average PP100C of the entire volume does not vary greatly with the stretching of pores.

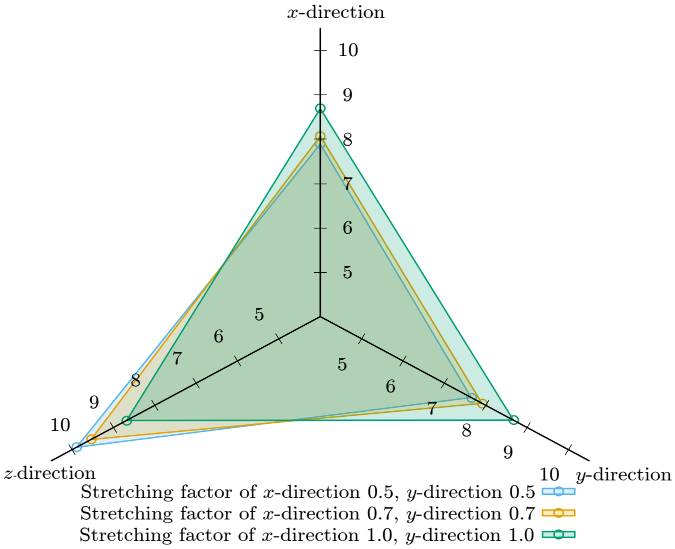
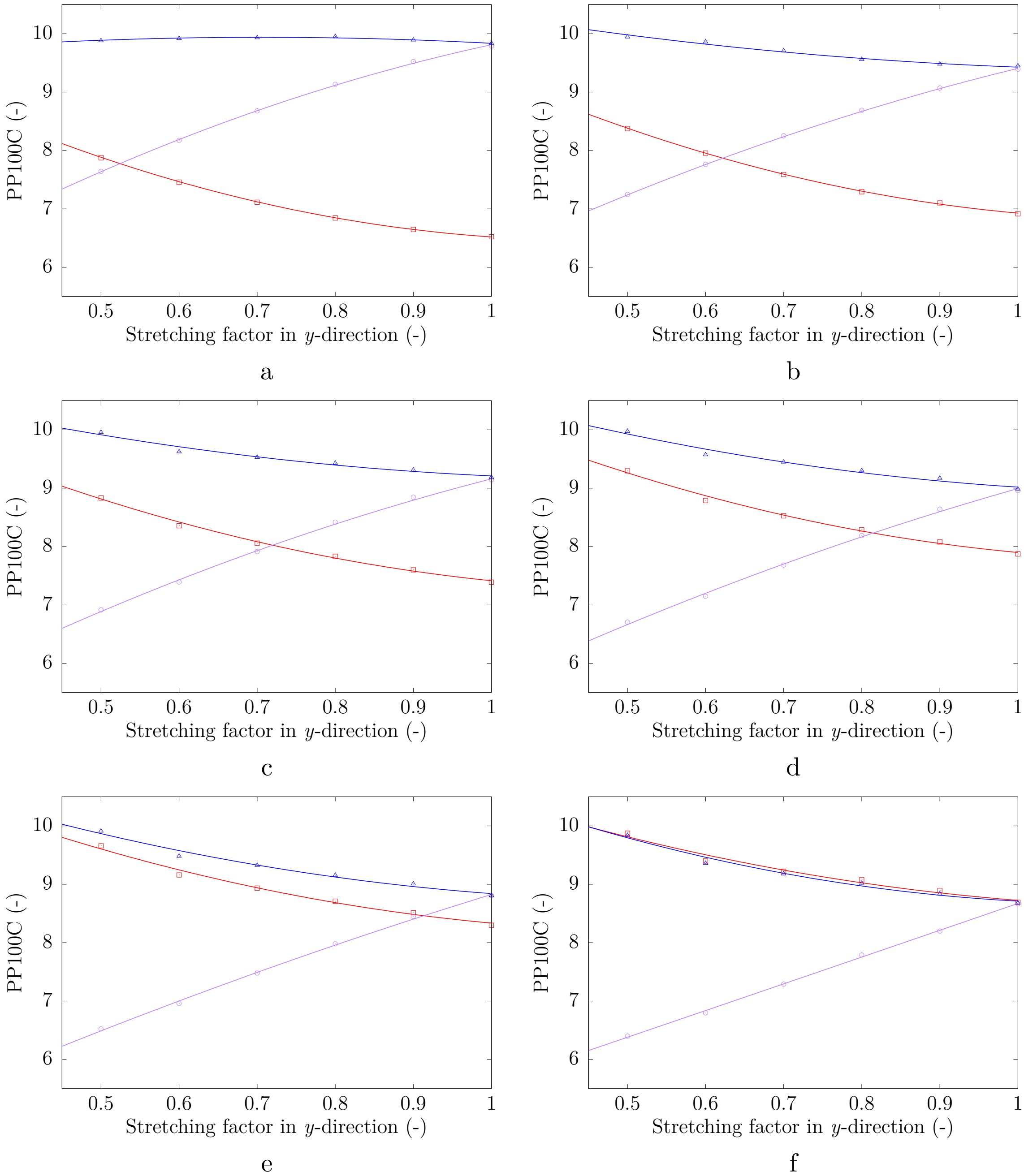


Figure 5.8.: The PP100C of structures stretched in two directions with factor 0.5 and 0.7.

Direction	Computed PP100C	Fit function
x-direction	□	—
y-direction	○	—
z-direction	△	—



a

b

c

d

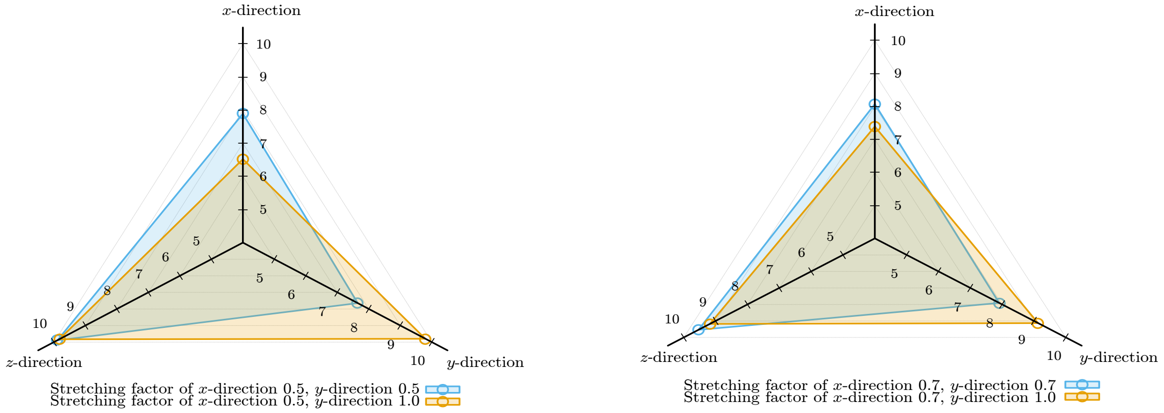
e

f

Figure 5.9.: The PP100C for structures with stretching in x and y-directions, where the stretching factor in x-direction is: (a) 0.5; (b) 0.6; (c) 0.7; (d) 0.8; (e) 0.9; (f) 1.0. Each point represents data from three structures with different seeds.

Comparison of structures with one and two stretching directions

As shown in Figure 5.10, if stretching can only be found in the x-direction of the structures (sandy brown), the PP100C computed in this direction is much smaller than that in the other two directions and is influenced by the stretching factor deeply. When stretching also occurs in the y-direction (blue), the computed PP100C in the x-direction becomes larger, while that in the y-direction becomes smaller and changes only slightly in the remaining direction. The greater the degree of stretching, the more pronounced this phenomenon is.



(a)

(b)

Figure 5.10.: The PP100C of structures with one stretching direction (sandy brown) and two stretching directions (blue), where the stretching factor in x-direction is: (a) 0.5; (b) 0.7.

5.2. PPI for real structures

The computed PP100C needs to be converted to obtain PPI. The real diameter of the specimen is known to be 4 cm [14]. In the pore network model, the domain in the x-direction is considered to represent the diameter of the foam. The conversion equation is as follows

$$\mathrm{PPI} = \frac{2.54}{4} \cdot C_n \cdot \frac{\mathrm{PP100C}}{100}\,, \tag{5.1}$$

where C_n represents the number of cells of the domain in x-direction, and the results are also computed using two approaches. Table 5.7 shows the results computed by the original algorithm. It can be seen that the PPI values are smaller than those given by the manufacturer and the difference is more pronounced in the 30 PPI structure. Table 5.8 presents the results computed by the improved algorithm. Although the results are closer to the values given by the manufacturer than the original algorithm, they are still on the small side.

A special point is that the PPI of the z-direction computed by the original algorithm is bigger than the other two directions, but the results of the improved algorithm are opposite, the PPI of z-direction is the smallest. Figure 5.11 is the x-z cross-section images of the segmented models, where can be seen that the shape of the defined pores is closer to an ellipse and has a greater length in the z-direction,i.e., there are stretching of pore radius in the z-direction. The number of pores in the z-direction in one inch should be less. So the PPI value in the z-direction should also be smaller than the other two directions. The improved algorithm can produce more accurate results and the PPI value can, to a certain extent, express the anisotropy of a porous structure.

Table 5.7.: The PPI for real structures computed by the original algorithm.

Company provided PPI (-)	Computed PPI (-) in different directions			
	x-direction	y-direction	z-direction	arithmetic mean in all three directions
10	7.51	7.52	7.81	**7.61**
30	10.62	10.28	11.39	**10.76**

Table 5.8.: The PPI for real structures computed by the second algorithm.

Company provided PPI (-)	Computed PPI (-) in different directions			
	x-direction	y-direction	z-direction	arithmetic mean in all three directions
10	9.68	10.14	8.19	**9.34**
30	14.08	13.85	11.42	**13.12**

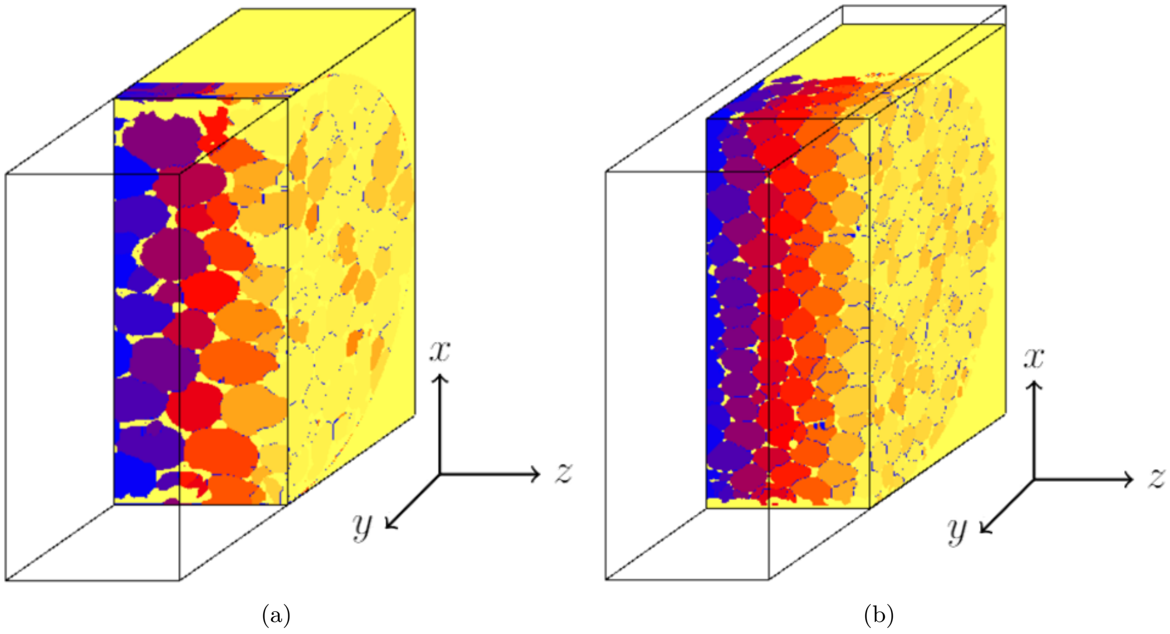


Figure 5.11.: The x-z cross section images of the segmented models: (a) 10 PPI structure; (b) 30 PPI structure.

6. Conclusion and future work

6.1. Conclusion

In this work, an algorithm is implemented to compute the PPI in the three spatial directions of a porous structure, the pore segmentation is based on a marker-based Watershed algorithm, which was proposed by Gostick et al. [17].

The accuracy of the original counting algorithm is verified using an aligned structure. Because the distribution of pores in porous structures is always diverse, the original algorithm is modified to work better with the complex structures generated in the PACE3Denvironment. Both of the modifications are designed to reduce the effect of periodic boundaries on the computed results and are also validated by a periodic aligned structure. The first modification is that the measured number of pores needs to be subtracted by one to exclude double-counted pores, which repeat at the boundary, the second modification is by adjusting the range of the measuring lines. It is concluded that the results of the first modified algorithm are more indicative of the variation trend in the PPI values (see Figure 5.1), while the second modified algorithm gives more precise results for higher porosities, larger numbers of Voronoi points (pores) structures (see Figure 5.2). For structures with more Voronoi points and high porosities (see Table 5.3), the difference between the two modified algorithms is smaller.

The stretching of pores has effects on the mechanical properties and the permeation of liquids in porous structures. Porous structures with stretched pores in targeted directions are also common in industrial production. Therefore, the relationship between PPI and the stretching of pores is noteworthy. When the pores are stretched in one direction, the computed PPI in that direction becomes smaller, the computed PPI in the other directions increases (see Figure 5.7). With a higher degree of stretching (in this thesis is a smaller stretching factor), the greater the variation of the computed PPI in all spatial directions. When there is no stretching in the structures, the computed PPI in the three spatial directions is identical. When the stretching of pores can be found in two directions, the PPI in both directions is lower than in the remaining direction. When the degree of stretching is the same in both directions, the computed PPI is approximately the same (see Figure 5.9). When the conditions of structures generation are constant, the direction and degree of stretching of the pores do not change the average PPI in the entire volume significantly.

The general rules of the variation of PPI are as follows: For the structures with the same porosity, those with higher numbers of Voronoi points (pores) have higher PPI values. When the structures have the same number of Voronoi points, PPI increase with increasing porosity. Furthermore, the structures with more Voronoi points are more sensitive to the variation of strut radius. As the strut radius increases, the PPI changes sharply in structures with a high number of Voronoi points, while the PPI in structures with fewer Voronoi points changes gently (see Figure 5.4 and Figure 5.5). Because the porosity of structures with more

Voronoi points changes a lot when the strut radius varies. With an increment of strut radius, the structures have a smaller PPI. In addition, as the stretching factor increases (degree of stretching decreases), the PPI computed in this direction becomes larger and the PPI in the other directions becomes smaller.

The end of the work is applying the algorithm to the real structures. First, the structure is reconstructed from CT images and then PPI values are computed. It can be found that the second modified algorithm, which is improved by adjusting the length of the measuring line, produces results closer to those given by the manufacturer. The results are still less than the data given by the manufacturer, especially for structures with large PPI values. At the same time, it is also found that PPI values can demonstrate partly the anisotropy of the structures. The PPI in the z-direction is smaller than the other directions, from the cross-section, it can be found that the pores are indeed stretched along the z-direction.

6.2. Future work

In the analysis of the results in chapter 5, it is proposed that there is a critical range of strut radius. This issue is essentially a question of what has the greater influence on the PPI value, the number of Voronoi points or the porosity. As can be seen in Table 5.5, when the strut radius is 1, the difference between the pore numbers of the individual structures (from 250 to 100, around 60% difference) is much greater than the difference between the porosities (from 88.25% to 93.3%), and now the number of pores play a decisive role and influences the PPI values. However, when the strut radius is 4, the difference between the number of pores remains the same and the difference in porosity becomes progressively greater (from 23.52% to 46.3%). This difference is still smaller than the difference between the number of pores, but at this time, the changes of PPI do not follow the previous rules. Does this mean that the number of pores has a much smaller effect on PPI than porosity?

Subsequent works can use the above option as an entry point to find the magnitude of the effect of pore numbers and porosity on PPI. At the same time, the position of the measuring lines can be further improved, for example by not placing measuring lines in the same direction as the axes, but at an angle to them.

A. List of symbols

Latin letters

C_n	the number of cells of the domain in x-direction, $[\mathrm{m}^{-1}]$
C_t	tested number of pores
d_edge	thickness of ligament, [m]
d_face	diameter of windows, [m]
d_pore	diameter of pore, [m]
$d(\mathbf{x}, \mathbf{s})$	Euclidean distance
l_m	the length of the measuring line, $[\mathrm{m}^{-1}]$
l_edge	length of ligament, [m]
N_p	the number of pores
S_V	specific surface area, $[\mathrm{m}^{-1}]$
V_v	volume of void space, $[\mathrm{m}^3]$
V_t	total volume, $[\mathrm{m}^3]$

Greek letters

ε	porosity, $[-]$
Ω	Euclidean space
Ω^+	pore space
Ω^-	structure space
$\partial_s\Omega$	surface space

Abbreviation

Pace3D	Parallel Algorithms for Crystal Evolution in 3D
PNM	pore network model
PP100C	pores per 100 cells
PPI	pores per inch

B. List of codes

B.1. The original algorithm

```c
/***************************************************************
 *
 *   @pace3d     Parallel Algorithms for Crystal Evolution in 3D
 *
 *   @copyright  2022 IDM - Institute of Digital Materials @n
 *                    University of Applied Sciences @n
 *                    Karlsruhe @n
 *
 *   @file       ppiWatershed_neu.c
 *   @ingroup    analyser
 *   @brief      This is a tool for PPI-counting. It is used on the segmented images with marker-based watershed algorithm.
 *   @todo       The number of pores can be counted.
 *
 ***************************************************************
 *
 *   @lastmodified Tianyu LU 07.02.2022
 *
 *************************************************************/
#include <wrapper.h>
#include <datafiles/fielddescriptor.h>
#include <container/queue.h>
#include <container/datavector.h>
#include <container/genericfields.h>
#include <timing/timer.h>
#include <plot/table.h>

static frames_t frameList = {.frames = NULL, .count = 0};

static char *fileIn;
static long force = false;
static plot_t      plotter;

argument_t arguments[] = {
 {NULL, ' ', PARAM_REQUIRED, "", NULL,    "image (vtk, png) or SimGeo (generated by skeletonize) to process", PARAM_FILEIN,  &fileIn},
};

toolparam_t tool = {
 .description   = "This tool processes....\n",
 .example       = "e.g.: .",
 .istested      = 0,
 .arguments     = arguments
};

float PoreNumber(float store[],long len,long Max,long Min) { //function for counting every pore on the measuring lines
 float count=0;
 long tmpi=0;
 long tmpj=0;
 if (store[0]>Min && store[0]!=Max) {
 count=count+1;
 }
 for(tmpi=1;tmpi<len;tmpi++) {
  for(tmpj=1;tmpj<=tmpi;tmpj++) {
   if(store[tmpi]!=store[tmpi-tmpj] && store[tmpi]>Min && store[tmpi]!=Max && tmpj==tmpi) {
    count++;
    break;
   } else if(store[tmpi]!=store[tmpi-tmpj] && store[tmpi]>Min && store[tmpi]!=Max && tmpj!=tmpi) {
    count = count + 0;
    } else {
       break;
     }
   }
 }
 return (float)count;
}

float my_Average(long a,long b) {
 return (float)a/b;
}

float VolumPpi( SimBench *in,ScalarData *b,char a,int num) { //count the pore in different directions
 long dir_x;
 long dir_y;
 long dir_z;
 ulong x_max=in->simgeo->Nx;//get the domain of structures
```

```
  ulong z_max=in->simgeo->Nz;
  ulong y_max= in->simgeo->Ny;
  long lenz=z_max;
  float tmp_storez[lenz];//change with domain
  long lenx=x_max;
  float tmp_storex[lenx];
  long leny=y_max;
  float tmp_storey[leny];
  int init=0;
  long c_num=0;
  float sum_vx=0;
  long times_vx=0;
  long times_px=0;
  float sum_vy=0;
  long times_vy=0;
  long times_py=0;
  float sum_vz=0;
  long times_vz=0;
  long times_pz=0;
  long times_np=0;
  float min, max, min_i, max_i;
  ScalarData_getMinMax(b, &min, &max, &min_i, &max_i);
  if(a=='x') {// x-direction
    for(dir_y=0;dir_y<y_max;dir_y=dir_y+num) {
      times_px=0;
      times_vx++;
      for(dir_z=0;dir_z<z_max;dir_z=dir_z+num) {
        times_px++;
        for(init=0;init<x_max;init++) {
        tmp_storex[init]=-1.0;
        }
        for(dir_x=0;dir_x<x_max;dir_x++) {
          float test = ScalarData_getValue(b, dir_x, dir_y,
dir_z);
          if(test>=min && test<=max) {
              tmp_storex[c_num]=test;
              c_num++;
          }
        }
        long length_total=c_num;
        c_num=0;
        float result =PoreNumber(tmp_storex,length_total,max,min
) ;
        if(result==0) {
          times_np++;
        }
        sum_vx=sum_vx+result;
      }
    }
    float time_totalx = times_vx * times_px-times_np;
    float average_v=my_Average(sum_vx,time_totalx)*100/lenx; //transform the output results into PP100C
    return average_v;
  }
  else if(a=='y') {//y-direction
    for(dir_z=0;dir_z<z_max;dir_z=dir_z+num) {
      times_py=0;
      times_vy++;
      for(dir_x=0;dir_x<x_max;dir_x=dir_x+num) {
        times_py++;
        for(init=0;init<y_max;init++) {
          tmp_storey[init]=-1.0;
        }
        for(dir_y=0;dir_y<y_max;dir_y++) {
          float test = ScalarData_getValue(b, dir_x, dir_y, dir_z);
          if(test>=min && test<=max) {
            tmp_storey[c_num]=test;
            c_num++;
          }
        }
        long length_total=c_num;
            c_num=0;
            float result =PoreNumber(tmp_storey,length_total,max,min) ;
            if(result==0) {
          times_np++;
            }
            sum_vy=sum_vy+result;
      }
    }
    float time_totaly = times_vy * times_py-times_np;
    float average_v=my_Average(sum_vy,time_totaly)*100/leny;
    return average_v;
  } else {//z-direction
    for(dir_y=0;dir_y<y_max;dir_y=dir_y+num) {
      times_vz++;
      times_pz=0;
      for(dir_x=0;dir_x<x_max;dir_x=dir_x+num) {
        times_pz++;
        for(init=0;init<z_max;init++) {
          tmp_storez[init]=-1.0;
        }
        for(dir_z=0;dir_z<z_max;dir_z++) {
          float test = ScalarData_getValue(b, dir_x, dir_y, dir_z);
          if(test>=min && test<=max) {
                tmp_storez[c_num]=test;
                c_num++;
          }
        }
        long length_total=c_num;
        c_num=0;
        float result =PoreNumber(tmp_storez,length_total,max,min) ;
        if(result==0) {
          times_np++;
        }
        sum_vz=sum_vz+result;
      }
    }
    float time_totalz = times_vz * times_pz-times_np;
    float average_v=my_Average(sum_vz,time_totalz)*100/lenz;
    return average_v;
    }
}


int main(int argc, char *argv[])
{
  PACE3DMAIN(argv[0]);
  getParams(argc, argv, tool, ARGUMENT(arguments));

  SimBench *in = newSimBench();
  ScalarData *combinedImageData = NULL;
  Table *tb = newTable();
  // IN
  SimBench_init(in, fileIn, SIMBENCH_IN);
  SimBench_addFileByFieldname(in, "combined", FD_SCALARDATA);
  SimBench_finishInit(in, force);
  SimBench_getFirstFrame(in, &frameList, false); // actually loads the scalar data into "in->scalardata[0]"
  combinedImageData = in->scalardata[0];
  float average_vx =  VolumPpi( in,combinedImageData,'x', 1);
  float average_vy =  VolumPpi( in,combinedImageData,'y', 1);
  float average_vz =  VolumPpi( in,combinedImageData,'z', 1);
  Table_init(tb, &plotter, " l f f  f  f", "frame", &in->frame, "time", in->simulationtime,"PP100C_x", &average_vx, "PP100C_y", &average_vy,"PP100C_z", &average_vz);
  Table_printRow(tb);
}
```

B.2. The improved parts

B.2.1. The improved parts of the first algorithm

```
if(a=='x') {
    for(dir_y=0;dir_y<y_max;dir_y=dir_y+num) {
      times_px=0;
      times_vx++;
      for(dir_z=0;dir_z<z_max;dir_z=dir_z+num) {
        times_px++;
        for(init=0;init<x_max;init++) {
        tmp_storex[init]=-1.0;
        }
        for(dir_x=0;dir_x<x_max;dir_x++) {
          float test = ScalarData_getValue(b, dir_x, dir_y, dir_z);
          if(test>=min && test<=max) {
              tmp_storex[c_num]=test;
              c_num++;
          }
        }
        long length_total=c_num;
        long test_begin=tmp_storex[0];         //get the index of the first cell
        long test_end=tmp_storex[c_num-1]; //get the index of the last cell
        c_num=0;
        long result =PoreNumber(tmp_storex,length_total,max,min);
        if(result==0) {
          times_np++;
        }
        if(test0>min && test0<max && test_end>min && test_end<max){// determine if there are pores on the boundary.
        result=result-1;//if there are pores on the boundary, subtract the duplicate one
        }
        sum_vx=sum_vx+result;
      }
    }
    float time_totalx = times_vx * times_px-times_np;
    float average_v=my_Average(sum_vx,time_totalx)*100/lenx;
    return average_v;
  }
```

B.2.2. The functions for adjusting measuring lines

```
long Line_Begin(float store[],long len,long Min,long Max){//Find the beginning of the measuring lines.
  long i=0;
  long j=0;
  long count=0;
  if (store[0]>Min && store[0]!=Max) {
    count=count+1;
   }
  for(i=1;i<len;i++) {
    for(j=1;j<=i;j++) {
          if(store[i]!=store[i-j] && store[i]>0 && store[i]<Max && j==i) {
            count++;break;
          }
         else if(store[i]!=store[i-j] && store[i]>0 && store[i]!=Max && j!=i) {
           count = count + 0;
         }
         else
           break;
    }
    if(count==2) {//The first cell of the second pore has been found.
      break;
    }
  }
  if(i==len)
    return len-1;//The second pore cannot be fonud, set the beginning of the measuring lines at the last cell.
  else
    return i;// output the location of the frist cell of the second pore.
}


long Line_End(float store[],long len,long Min,long Max) {//Finde the end of the measuring lines.
  long i;
  long j;
  long count=0;
  if (store[len-1]>Min && store[len-1]!=Max) {
    count=count+1;
  }
  for (i=len-2;i>=0;i--) {
    for(j=1;j<=len-i;j++) {
          if(store[i]!=store[i+j] && store[i]>0 && store[i]<Max && j==len-i) {
```

```
      count++;break;
     }
     else if(store[i]!=store[i+j] && store[i]>0 && store[i]!=Max && j!=len-i) {
      count = count + 0;
     }
     else
      break;
  }
    if(count==2) {
     break;
    }
 }
 if(i==-1)// The penultimate pore cannot be found, set the end of the measuring line at the first cell.
  return 0;
 else
  return i;// output the location of the penltimate pore.
}
float get_result(long x,long y,long begin,long end,long total,long Min,long Max,float store[]) { // This function is used to calculate the length of the measuring lines and output pp100c.
 float result=0;                              //x y is the index of the first and last cell, only when there are both pores
 long length=0;                               // the length of measuring lines should be changed.
 if(x>Min && x<Max && y>Min && y<Max) {// when the first cell and the last cell are pores
  if(begin>end) { // no pores on the measuring lines
     result=0;
    }
    else {
     length=end-begin+1; //calculate the length of measuring lines.
     result = PpiWatershed_poreNumber(store,begin,end,length,Min,Max);//call the function to calculate the pp100c
    }
 }
 else {// when the first cell and the last cell are not both pores, the length of measuring lines is the domain.
  length=total;
  result =PpiWatershed_poreNumber(store,0,total,length,Min,Max);
 }
 return result;
```

```
}
float PpiWatershed_poreNumber(float store[],long begin,long end,long length,long Min,long Max) { // count every pore of the measuring lines and output in form pp100c.
  float count=0;
  long tmpi=0;
  long tmpj=0;
  if (store[begin]>Min && store[begin]<Max) {
  count=count+1;
  }
  for(tmpi= begin+1;tmpi<end;tmpi++) {
        for(tmpj=begin+1;tmpj<=tmpi;tmpj++) {
          if(store[tmpi]!=store[tmpi-tmpj+begin] && store[tmpi]>Min && store[tmpi]<Max && tmpj==tmpi) {
            count++;break;
          }
          else if(store[tmpi]!=store[tmpi-tmpj+begin] && store[tmpi]>Min && store[tmpi]<Max && tmpj!=tmpi) {
            count = count + 0;
          }
          else
            break;
        }
  }
  return (float)count*100/length;
}
```

Bibliography

[1] Anwar S. Al-Kharusi and Martin J. Blunt. Network extraction from sandstone and carbonate pore space images. *Journal of petroleum science and engineering*, 56(4):219–231, 2007.

[2] Patrick Altschuh. *Skalenübergreifende Analyse makroporöser Membranen im Kontext digitaler Zwillinge*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2020.

[3] Patrick Altschuh, Yuksel C. Yabansu, Johannes Hötzer, Michael Selzer, Britta Nestler, and Surya R Kalidindi. Data science approaches for microstructure quantification and feature identification in porous membranes. *Journal of Membrane Science*, 540:88–97, 2017.

[4] Anastasia August, Jörg Ettrich, Michael Rölle, Stefan Schmid, Marco Berghoff, Michael Selzer, and Britta Nestler. Prediction of heat conduction in open-cell foams via the diffuse interface representation of the phase-field method. *International Journal of Heat and Mass Transfer*, 84:800–808, 2015.

[5] Franz Aurenhammer. Voronoi diagramsa survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[6] Christopher A. Baldwin, Andrew J. Sederman, Michael D. Mantle, Paul Alexander, and Lynn F. Gladden. Determination and characterization of the structure of a pore space from 3d volume images. *Journal of colloid and interface science*, 181(1):79–92, 1996.

[7] Martin Beer, Radim Rybar, and Dušan Kudelas. Method of determining the essential geometric characteristics of the metal foam structure. *Metalurgija*, 58(3-4):211–214, 2019.

[8] Serge Beucher. The watershed transformation applied to image segmentation. *Scanning Microscopy*, 1992(6):28, 1992.

[9] Pradeep Bhattad, Clinton S. Willson, and Karsten E. Thompson. Effect of network structure on characterization and flow modeling using x-ray micro-tomography images of granular and fibrous porous media. *Transport in Porous Media*, 90(2):363–391, 2011.

[10] Florina C. Buciuman and Bettina Kraushaar-Czarnetzki. Ceramic foam monoliths as catalyst carriers. 1. adjustment and description of the morphology. *Industrial & engineering chemistry research*, 42(9):1863–1869, 2003.

[11] Thomas Byholm, Martti Toivakka, and Jan Westerholm. The application of morphological algorithms on 3-dimensional porous structures for identifying pores and gathering statistical data. *WSEAS Transactions on Information Science and Applications*, 3(12):2374–2380, 2006.

[12] Hu Dong and Martin J. Blunt. Pore-network extraction from micro-computerized-tomography images. *Physical review E*, 80(3):036307, 2009.

[13] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM review*, 41(4):637–676, 1999.

[14] Jörg Ettrich. *Fluid flow and heat transfer in cellular solids*, volume 39. KIT Scientific Publishing, 2014.

[15] Adrian Fisher. Cloud and cloud-shadow detection in spot5 hrg imagery with automated morphological feature extraction. *Remote Sensing*, 6(1):776–800, 2014.

[16] Lorna J. Gibson. Cellular solids. *Mrs Bulletin*, 28(4):270–274, 2003.

[17] Jeff T. Gostick. Versatile and efficient pore network extraction method using marker-based watershed segmentation. *Physical review E*, 96(2):023307, 2017.

[18] Johannes Hötzer, Andreas Reiter, Henrik Hierl, Philipp Steinmetz, Michael Selzer, and Britta Nestler. The parallel multi-physics phase-field framework pace3d. *Journal of computational science*, 26:1–12, 2018.

[19] Zeyun Jiang, Kejian Wu, Gary Couples, Marinus Izaak Jan Van Dijke, Kenneth Stuart Sorbie, and Jingsheng Ma. Efficient extraction of networks from three-dimensional porous media. *Water resources research*, 43(12), 2007.

[20] Zohaib Atiq Khan, Tom Tranter, Mehrez Agnaou, Ali Elkamel, and Jeff Gostick. Dual network extraction algorithm to investigate multiple transport processes in porous materials: Image-based modeling of pore and grain scale processes. *Computers & Chemical Engineering*, 123:64–77, 2019.

[21] Stephen J. Kline. The Purposes of Uncertainty Analysis. *Journal of Fluids Engineering*, 107(2):153–160, 06 1985.

[22] Peter Klobes and Ronald G. Munro. Porosity and specific surface area measurements for solid materials, 2006-08-01 2006.

[23] Willfried Kunz. Simulative benetzungsuntersuchung in porösen membranstrukturen mithilfe der phasenfeldmethode. Master's thesis, Karlsruhe Institute of Technology, 2019.

[24] Matthieu Laqua. Entwicklung eines algorithmus zur Überführung einer offenporigen und hochporösen mikrostruktu in ein porennetzwerkmodell, 2020.

[25] Jian Lei, Baozhi Pan, and Lihua Zhang. Advance of microscopic flow simulation based on digital cores and pore network. *Progress in Geophysics*, 33(2):653–660, 2018.

[26] Adrian Ley, Patrick Altschuh, Volkmar Thom, Michael Selzer, Britta Nestler, and Philipp Vana. Characterization of a macro porous polymer membrane at micron-scale by confocal-laser-scanning microscopy and 3d image analysis. *Journal of membrane science*, 564:543–551, 2018.

[27] zhirong. Liang, Celso P. Fernandes, Fábio S. Magnani, and Paulo C. Philippi. A reconstruction technique for three-dimensional porous media using image analysis and fourier transforms. *Journal of Petroleum Science and Engineering*, 21(3-4):273–283, 1998.

[28] W. Brent Lindquist, Sang-Moon Lee, David A. Coker, Keith W. Jones, and Per Spanne. Medial axis analysis of void structure in three-dimensional tomographic images of porous media. *Journal of Geophysical Research: Solid Earth*, 101(B4):8297–8310, 1996.

[29] P. S. Liu and Guo Feng Chen. *Porous materials: processing and applications.* Elsevier, 2014.

[30] Fereidoun Rezanezhad, William L. Quinton, Jonathan S. Price, David Elrick, Thomas R. Elliot, and Richard J. Heck. Examining the effect of pore size distribution and shape on flow through unsaturated peat using computed tomography. *Hydrology and Earth System Sciences*, 13(10):1993–2002, 2009.

[31] Jean Rouquerol, David Avnir, Craig W. Fairbridge, Douglas H. Everett, J. M. Haynes, Nicola Pernicone, John D. F. Ramsay, Kenneth Stafford William Sing, and Klaus K. Unger. Recommendations for the characterization of porous solids (technical report). *Pure and applied chemistry*, 66(8):1739–1758, 1994.

[32] Michael Röller. Füllalgorithmen zur generierung schaumartiger strukturen in 3d, 2010.

[33] Muhammad Sahimi. *Flow and transport in porous media and fractured rock: from classical methods to modern approaches.* John Wiley & Sons, 2011.

[34] Michael Scheffler and Paolo Colombo. *Cellular ceramics: structure, manufacturing, properties and applications.* John Wiley & Sons, 2006.

[35] Qingrong Xiong, Todor G. Baychev, and Andrey P. Jivkov. Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport. *Journal of contaminant hydrology*, 192:101–117, 2016.