TORT3D: A MATLAB code to compute geometric tortuosity from 3D images of unconsolidated porous media

Riyadh I. Al-Raoush a,⁎, Iman T. Madhoun b

a Department of Civil and Architectural Engineering, Qatar University, PO Box 2713, Doha, Qatar
b Environmental Engineering Master Program, College of Engineering, Qatar University, PO Box 2713, Doha, Qatar

Abstract

Tortuosity is a parameter that plays a significant role in the characterization of complex porous media systems and it has a significant impact on many engineering and environmental processes and applications. Flow in porous media, diffusion of gases in complex pore structures and membrane flux in water desalination are examples of the application of this important micro-scale parameter. In this paper, an algorithm was developed and implemented as a MATLAB code to compute tortuosity from three-dimensional images. The code reads a segmented image and finds all possible tortuous paths required to compute tortuosity. The code is user-friendly, easy to use and computationally efficient, as it requires a relatively short time to identify all possible connected paths between two boundaries of large images. The main idea of the developed algorithm is that it conducts a guided search for connected paths in the void space of the image utilizing the medial surface of the void space. Once all connected paths are identified in a specific direction, the average of all connected paths in that direction is used to compute tortuosity.

Three-dimensional images of sand systems acquired using X-ray computed tomography were used to validate the algorithm. Tortuosity values were computed from three-dimensional images of nine different natural sand systems using the developed algorithm and compared with predicted values by models available in the literature. Findings indicate that the code can successfully compute tortuosity for any unconsolidated porous system irrespective of the shape (i.e., geometry) of particles.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Tortuosity is a significant parameter that has a profound impact on flow and transport characteristics of porous media. Tortuosity is defined as the average ratio of the lengths of the actual paths that molecules travel from one point to the final destination through the pore (void) space of a porous medium to the length of the straight path between these points [58]. The importance of tortuosity can be observed in many engineering applications (e.g., diffusion, mass transfer and separation of mixtures) [10,16,40,42,68]. In geosciences applications, tortuosity impacts flux of the fluids in the pore space [28,72,74,89]. In acoustics applications, tortuosity is a significant parameter that impacts sound propagation and its velocity variations in the pore space [3,42,90]. In water applications, membrane tortuosity is a major factor that affects its distillation performance [35].

Tortuosity has a significant impact on single and multiphase flow and transport properties [7,46,60,62]. For instance, diffusion and permeability of fluids in a porous medium are directly related to its tortuosity. Diffusion coefficient can be estimated as [27]:

\[ D_p = \frac{\varepsilon D_b}{\tau^2} \]  

(1)

where \( D_p \) is the diffusion coefficient in the porous media \([L^2 T^{-1}]\), \( D_b \) is the diffusion coefficient in air or water, \( \varepsilon \) is porosity and \( \tau \) is tortuosity.

Whereas permeability, \( k \), is related to tortuosity, \( \tau \), by Kozeny-Carman equation as [85]:

\[ k = \frac{\varepsilon^2}{\beta^2 \tau^2 S^2} \]  

(2)

where \( \varepsilon \) is porosity, \( \beta \) is a shape related factor, and \( S \) is the average pore perimeter [85].

Moreover, tortuosity of a porous medium influences the magnitude and detailed structure of the entrapped fluids in the pore space [6,73]. The distribution and configurations of the entrapped fluids in the void space of subsurface systems is critical to enhanced oil recovery, rate and efficiency of hydrocarbon production, carbon sequestration...
management [23,51,77,84] and the integrity of subsurface storage applications [15,32,87].

Geometric tortuosity, $\tau_g$, can be defined as [25]:

$$\tau_g = \frac{<L_g>}{L_s}$$  \(3\)

where $<L_g>$ is the average length of true paths through the porous media and $L_s$ is the length of the straight-line across the porous media in the direction of flow. Geometric tortuosity can also be defined as the ratio of the shortest pathway to the straight-line length [1]. Based on the definition given in Eq. (3), the value of geometric tortuosity is always greater than one, whereas the coefficient of geometric tortuosity, ($\Gamma_g = \frac{1}{\tau_g}$), is less than one [25].

Several analytical, experimental, and numerical approaches have been used to determine tortuosity of porous media. Ahmadi et al. [2] proposed analytical expressions of tortuosity and permeability using the concept of representative elementary volume (REV) of cubic array of spheres. Lanfrey et al. [41] and Li and Yu [43] numerically modeled the porous media systems. Most imaged-based algorithms are implemented in commercial softwares that are not readily available to researchers as they are very costly. Moreover, diffusion-based experiments typically used for tortuosity measurements, fail to link tortuosity to the geometry and topology of the pore space.

Therefore, given the significance of tortuosity and its impact on many engineering applications coupled with the advances in the use of 3D imaging technologies, there is a need to develop an accurate, efficient and accessible code that can be used to obtain tortuosity of unconsolidated porous media from 3D images. The objective of this paper is to present, TORT3D, a MATLAB code that was developed to compute geometric tortuosity from 3D images. The code not only provides a scalar value of tortuosity, but also provides a three-dimensional description of all connected paths in the image, which in turn links tortuosity to the geometry, and topology of the pore space (i.e., connectivity, pore-body and pore-throat distributions). Such link is critical for three-dimensional simulations of flow and transport phenomena in porous media. The code reads segmented (i.e., binary) images and can be applied to 3D images of any unconsolidated porous media systems to identify all possible tortuous paths. TORT3D is available to the scientific community in the Supplementary Material section of the Journal. The paper demonstrates the capability of the code to compute tortuosity from images of different natural sand systems.

The paper is organized as follows: Section 2 presents the main steps of the algorithm, Section 3 provides application of the code to 3D Computed Tomography images of natural sand systems and comparison with existing tortuosity models and Section 4 presents conclusions.

2. Description of Tort3D code

The key functionality of Tort3D code is its capability to compute geometric tortuosity from 3D images of porous media using MATLAB environment. The code is user-friendly and easy to use where input parameters and user interaction are minimized. It can be used to compute tortuosity from 2D or 3D images. It has the option to select the direction of tortuosity computations along a given direction (i.e., x, y, or z) and the connectivity of voxels (i.e., number of neighboring voxels/pixels while searching for connected paths). Moreover, it is computationally efficient, as it requires a relatively short time (few minutes) to identify all possible connected paths between two boundaries of large images that are used to compute tortuosity. This is mainly due to the optimized coding scheme used where loops, nested loops, and “if statements” are limited in the code. Input and output variables are given in Table 1.

The general flow chart of the algorithm is given in Figs. 1 and 2. The algorithm commences by reading segmented (i.e., binary) images. Note that the segmentation process is beyond the scope of this paper, many published papers have presented and discussed different segmentation algorithms such as k-means clustering [37,39,45,86], Mean-Shift Clustering [19,22,53], watershed [26,36,47] and Normalized Graph Cuts [5,31,82].

The main idea of the developed algorithm is that it conducts a guided search for connected paths in the void space of the image utilizing the geometry.
medial surface of the void space. The medial surface is a three-dimensional surface that simplifies the void space and is constructed by stacking the two-dimensional medial axis of each slice. The advantage of this approach is that it limits the search along the medial surface, which in turn minimizes CPU time and memory requirements to find possible paths in the image. Once all connected paths are identified for a specific direction, the average of all connected paths in that direction is used to compute tortuosity. A connected path is defined as the one that starts from the first slice of the image and ends at the final slice of the image in the direction computation (i.e., connects image boundaries in the direction of flow). Tortuosity is then computed as:

$$
\tau = \frac{\sum_{i=1}^{n} l_i}{n}
$$

where \(l_i\) is a given path through the void space that connects the boundary of the images in the direction of flow, \(l_i\) is the corresponding straight line (i.e., size of the image in the direction of flow) and \(n\) is the number of paths. A brief description of the main elements of the code is described in the subsections below.

2.1. Generation of the medial surface of the void space

Three-dimensional medial surface of the void space is generated from two-dimensional skeletonization of the void space of each slice in the image. The medial surface guides the search for connected paths in the void space starting from predefined locations in the first slice in the direction of flow (more details are provided in the next section).

The medial surface provides an efficient representation of the void space by reducing its dimensionality while preserving its topology and geometry [71]. The MATLAB command “bwmorph (BW, ‘skel’, Inf)” is implemented to obtain the two-dimensional skeletonization for each slice. A controlled erosion is performed and repeated until the thickness of the skeleton of the void space becomes one voxel (i.e., no change is observed in the skeleton).

2.2. Initiating starting points (pixels) for the search for connected paths

While each pixel in the void space in the first slice in the direction of flow is a potential starting point for the search of connected paths, only those pixels that form junctions at the medial axis lead to non-redundant paths. These junctions form pore-bodies in the void space and represent centers of inscribed circles (shown as green circles in Fig. 3). To implement a computationally efficient algorithm, the search for connected paths is limited to those locations (i.e., centers of pore-bodies in the void space). Therefore, the main goal of this step in the algorithm is to find pixels in the first slice in the direction of flow (i.e., tortuosity computation) that serve as starting points for the search for connected paths.
2.3. Search for connected paths

Identifying connected paths along the three-dimensional medial surface depends on the connectivity of voxels (volume element in 3D images). The algorithm defines the connectivity of a given voxel (or pixel in 2D images) by identifying its neighboring voxels that connect through a face, edge or corner. In 2D images, there are 4 or 8 neighboring pixels for a given pixel, whereas there are 6, 18 or 26 neighboring voxels for a given 3D voxel. In the developed algorithm, the connectivity is defined before the search for connected paths starts. The main elements of the algorithm of finding connected paths are given in the flow chart shown in Fig. 2. The search for connected paths starts from the voxels saved in “Starting_Path_Index” variable that includes all possible starting points in the first slice of the direction of flow. Neighboring voxels of the selected voxel are determined and only voxels that belong to the medial surface are considered as potential voxels in the connected path. The voxel that represents the centroid of the voxels with maximum coordinate in the direction of flow is considered as a voxel in the connected path (i.e., “Next_Move_Index”). However, if the voxel (Next_Move_Index) belongs to a solid phase, it will be removed from consideration and the previous voxel is considered. Then, the algorithm identifies new neighboring voxels and repeats the search on the medial surface as shown in Fig. 2. The search continues until a connected path is found (i.e., path that connects two opposite faces of the image). The above procedure is repeated for all starting voxels saved in “Starting_Path_Index” variable.

3. Computation of tortuosity using the developed code

3.1. Three-dimensional images using X-ray computed tomography

In order to verify tortuosity computations of the developed code, 3D X-ray computed tomography images of different sand systems were used and tortuosity values were compared to values obtained from numerical and analytical models available in the literature. X-ray computed tomography is a powerful technique used to visualize the internal structure of porous media nondestructively. A three-dimensional image of an object is obtained by converting X-ray attenuation data to a series of cross-sections using image reconstruction algorithms while the sample is rotated by a specific angle. During sample rotation, the X-ray source produces an X-ray beam that passes through a section of the sample where detectors convert the X-ray to tomographs to create an image. During one complete rotation, tomographs are collected and converted to cross-sectional images (slices) of the sample (tomographic reconstruction). Three-dimensional images are then generated from a series of 2D images obtained while the sample is rotated around an axis of rotation.

In this work, 3D images of sand systems were acquired using X-ray computed tomography at the GeoSoilEnviroCARS beamline (13-BM-D) at the Advanced Photon Source, Argonne National Laboratory, USA. Image reconstruction algorithms developed by GeoSoilEnviroCARS were used to convert X-ray attenuation data to cross-sections and then to 3D images. More details can be found in [4].
3.2. Natural porous media systems

Three-dimensional images of nine different sand systems were acquired using computed tomography and used to verify tortuosity values obtained using the developed algorithm. Five sand systems (silica) that represent rounded particles and four systems (mixture of equal masses of silica and quartz sands) that represent irregular particles were used. Physical properties of the systems and sizes of images are given in Table 2. All characteristics given in the table were obtained from 3D images using image-processing algorithms. More details about sample preparations, characterization and tomography imaging can be found in [4]. Samples were selected to provide a tool for code verification by measuring geometric tortuosity of different porous media systems with a range of porosity, median grain diameter and shape indices. Porosity of the silica sand systems (i.e., S1, S2, S3, S4 and S5) ranges between 0.32 and 0.38. Whereas porosity for the mixed samples (M1, M2, M3 and M4) ranges between 0.39 and 0.43. Porosity values of mixed sands are higher than values obtained in the silica sand due to difficulty of obtaining dense compactions in systems composed of non-spherical grains. Roundness values of silica sands indicate that the systems are composed of highly rounded grains compared to the mixed sands.

Fig. 4 shows selected cross-sections of silica and mixed sands from 3D tomography images. Two phases are identified in both images, grains (dark red) and void (dark blue). Image resolution is 9.6 μm/pixel in all directions and the size of all images used for analysis is 380 × 380 × 520 voxels.

Fig. 4. Cross section of silica sand (S1) (left) and mixed sand (M1) (right). Red color depicts sand particles and dark blue color represents the void space.

Fig. 5. Representative Elementary Volume (REV) analysis of the systems used for tortuosity computations.

<table>
<thead>
<tr>
<th>Sand</th>
<th>Porous media</th>
<th>Image size (voxels)</th>
<th>ε</th>
<th>d_{50} (mm)</th>
<th>C_{u}</th>
<th>C_{g}</th>
<th>S_{i}</th>
<th>R_{i}</th>
<th>S (mm²)</th>
<th>SSA (mm⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silica</td>
<td>S1</td>
<td>380 × 380 × 520</td>
<td>0.37</td>
<td>0.35</td>
<td>1.85</td>
<td>1.12</td>
<td>0.90</td>
<td>0.83</td>
<td>248.33</td>
<td>19.46</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>380 × 380 × 520</td>
<td>0.33</td>
<td>0.27</td>
<td>1.79</td>
<td>1.06</td>
<td>0.89</td>
<td>0.81</td>
<td>316.41</td>
<td>23.55</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>380 × 380 × 520</td>
<td>0.35</td>
<td>0.27</td>
<td>1.79</td>
<td>1.07</td>
<td>0.89</td>
<td>0.81</td>
<td>340.35</td>
<td>26.18</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>380 × 380 × 520</td>
<td>0.32</td>
<td>0.26</td>
<td>1.83</td>
<td>1.00</td>
<td>0.89</td>
<td>0.82</td>
<td>315.08</td>
<td>22.68</td>
</tr>
<tr>
<td></td>
<td>S5</td>
<td>380 × 380 × 520</td>
<td>0.38</td>
<td>0.20</td>
<td>1.52</td>
<td>1.09</td>
<td>0.88</td>
<td>0.81</td>
<td>450.68</td>
<td>36.00</td>
</tr>
<tr>
<td>Mixed</td>
<td>M1</td>
<td>380 × 380 × 520</td>
<td>0.40</td>
<td>0.30</td>
<td>2.46</td>
<td>1.06</td>
<td>0.84</td>
<td>0.76</td>
<td>274.31</td>
<td>22.83</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>380 × 380 × 520</td>
<td>0.40</td>
<td>0.25</td>
<td>1.96</td>
<td>1.09</td>
<td>0.85</td>
<td>0.77</td>
<td>341.00</td>
<td>28.68</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>380 × 380 × 520</td>
<td>0.39</td>
<td>0.23</td>
<td>1.89</td>
<td>1.03</td>
<td>0.86</td>
<td>0.77</td>
<td>345.07</td>
<td>28.47</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>380 × 380 × 520</td>
<td>0.43</td>
<td>0.19</td>
<td>1.59</td>
<td>1.06</td>
<td>0.85</td>
<td>0.77</td>
<td>434.91</td>
<td>39.25</td>
</tr>
</tbody>
</table>

ε: porosity, d_{50}: median grain diameter, C_{u}: uniformity coefficient = d_{60} / d_{10}, C_{g}: coefficient of gradation = d_{30} / (d_{60} * d_{10}), S_{i}: sphericity index, R_{i}: roundness index, S: surface area, SSA: specific surface area = surface area/volume.
520 voxels. All images of this size are considered representative elementary volume (REV) for porosity as shown in Fig. 5.

3.3. Computation of tortuosity

The algorithm was applied to compute tortuosity of natural sand systems given in Table 2. Computed values of tortuosity for all systems are given in Tables 4 and 5. Fig. 6 shows a cross-sectional image obtained from a 3D image to visualize the accuracy of the algorithm to find connected paths in the image. As can be seen from the image, all tortuous paths were accurately identified in the direction of flow (i.e., from left to right of the image). Similarly, Fig. 7 shows a cross-sectional image obtained from a 3D image to visualize the connected paths in the image when the direction of flow is from top to bottom of the image. Note that all computations are in 3D, however, Figs. 6 and 7 are used to better visualize the connected paths. Fig. 8 shows all connected paths in a 3D image when the direction of flow is in the vertical direction.

Execution time required to compute tortuosity depends on the size of the image, complexity of the system analyzed and the machine used to perform the computations. Computational time needed to compute tortuosity was approximately 3 min for silica sands and 6 min for mixed sands using images with a size of 380 × 380 × 520 voxels using a laptop with a 2.5 GHz Intel Core i7 processor, with a memory of 16 GB RAM, 1600 MHz running under Windows 8.1 environment.

3.4. Comparison with available models

Tortuosity values of sand systems given in Table 2 were computed by the developed code and compared with tortuosity values predicted by models reported in the literature. Table 3 summarizes models reported in the literature and used for comparison along with their conditions, derivation method and references. Values of tortuosity computed using the developed code and predicted using the models reported in the literature and relative differences are listed in Tables 4 and 5.

As shown in Table 3, Models 1, 2, 3 and 4 were obtained from systems comprised of spherical particles where they relate tortuosity to porosity. Tortuosity values predicted by these models were compared to values computed using the developed code for similar systems (i.e., S1, S2, S3 and S4) as shown in Table 4. Findings indicate that there is a good agreement between tortuosity values obtained using Tort3D and those obtained from model predictions of similar systems.

As shown in Table 3, Model 5 relates tortuosity to porosity whereas Model 6 related tortuosity to the median grain diameter. Both models were obtained from systems comprised of irregular particles. Tortuosity values predicted by Models 5 and 6 were compared to values computed using the developed code for similar systems (i.e., M1, M2, M3 and M4) as shown in Table 5. Findings indicate that there is a good agreement between tortuosity values obtained using Tort3D and those obtained from model predictions of similar systems.

While tortuosity is expressed as scalar value, one of the main advantages of the developed algorithm is that it provides a three-dimensional description of the tortuous paths in the void space which can be related to properties of pore space and the solid particles. This in turn can be useful to understand many transport mechanisms at the pore space that are influenced by tortuosity.
Table 3

Tortuosity models available in the literature used to compare with tortuosity values obtained using the developed algorithm (i.e., Tort3D code).

<table>
<thead>
<tr>
<th>Model #</th>
<th>Model</th>
<th>Condition</th>
<th>Derivation method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\tau = 1 - 0.49 \ln \epsilon$</td>
<td>Bed of spheres</td>
<td>Experimental</td>
<td>[49]</td>
</tr>
<tr>
<td>2</td>
<td>$\tau = \frac{1}{\epsilon}$</td>
<td>Granular packings</td>
<td>Analytical</td>
<td>[8,21]</td>
</tr>
<tr>
<td>3</td>
<td>$\tau = 1 + 0.64 (1 - \epsilon)$</td>
<td>Spherical particles</td>
<td>Numerical</td>
<td>[63]</td>
</tr>
<tr>
<td>4</td>
<td>$\tau = \frac{\epsilon}{1 - (1 - \epsilon)^2}$</td>
<td>Isotropic granular media</td>
<td>Analytical (based on a representative unit cell)</td>
<td>[20]</td>
</tr>
<tr>
<td>5</td>
<td>$\tau = \sqrt{1 + 2(1 - \epsilon)}$</td>
<td>Sandy marine sediments</td>
<td>Experimental (diffusion experiment)</td>
<td>[30]</td>
</tr>
<tr>
<td>6</td>
<td>$\tau = 0.19 d_{50} + 1.45$</td>
<td>Equation derived for 3D computed tomography images of Granusil and Accusand sand packs</td>
<td>Medial Axis algorithm in 3DMA-Rock package</td>
<td>[54]</td>
</tr>
</tbody>
</table>

Table 4

Comparison between computed tortuosity of silica sand systems using Tort3D code with values obtained using Models 1, 2, 3 and 4 given in Table 3.

<table>
<thead>
<tr>
<th>Porous media</th>
<th>Tort3D Code</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>$\tau$</td>
<td>$\tau$</td>
<td>Relative difference</td>
<td>$\tau$</td>
<td>Relative difference</td>
</tr>
<tr>
<td>S2</td>
<td>1.46</td>
<td>1.49</td>
<td>1.59</td>
<td>1.39</td>
<td>5.16</td>
</tr>
<tr>
<td>S3</td>
<td>1.42</td>
<td>1.54</td>
<td>8.94</td>
<td>1.44</td>
<td>1.78</td>
</tr>
<tr>
<td>S4</td>
<td>1.43</td>
<td>1.51</td>
<td>5.85</td>
<td>1.41</td>
<td>1.17</td>
</tr>
<tr>
<td>S5</td>
<td>1.51</td>
<td>1.56</td>
<td>3.66</td>
<td>1.46</td>
<td>3.11</td>
</tr>
<tr>
<td>S6</td>
<td>1.41</td>
<td>1.48</td>
<td>4.72</td>
<td>1.38</td>
<td>2.24</td>
</tr>
</tbody>
</table>

4. Conclusions

In this paper, an algorithm was developed and implemented as a MATLAB code (i.e., Tort3D) to compute tortuosity from three-dimensional images. The code reads a segmented image and finds all possible tortuous paths that require computation tortuosity. The code is user-friendly and easy to use where input parameters and user interaction are minimized. It has the option to select the direction of tortuosity computations along a given direction (i.e., x, y, or z) and the connectivity of voxels (i.e., number of neighboring voxels/pixels while searching for connected paths). Moreover, it is computationally efficient, as it requires a relatively short time (few minutes) to identify all possible connected paths between two boundaries of large images. The main idea of the developed algorithm is that it conducts a guided search for connected paths in the void space of the image utilizing the medial surface of the void space. The advantage of this approach is that it limits the search along the medial surface, which in turn minimizes CPU time and memory requirements to find possible paths in the image. Once all connected paths are identified for a specific direction, the average of all connected paths in that direction is used to compute tortuosity.

Three-dimensional images of sand systems acquired using X-ray computed tomography were used to validate the code. Tortuosity was computed from three-dimensional images of nine different natural sand systems using the developed algorithm and compared with predicted values by models available in the literature. Findings indicate that the code can successfully compute tortuosity for any unconsolidated porous system irrespective of the shape (i.e., geometry) of particles.

Acknowledgment

Tomography imaging was performed at the GeoSoilEnviroCARS beamline (13-BM-D) at the Advanced Photon Source. GeoSoilEnviroCARS is supported by the National Science Foundation Earth Sciences (EAR-0217473), Department of Energy Geosciences (DE-FG01-94ER14466), and the state of Illinois. Use of the APS was supported by the U.S. Department of Energy, Basic Energy Sciences, Office of Energy Research, under contract No. W-31-109-Eng-38.

Appendix A. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.powtec.2017.06.066.

References


