

 Open access • Journal Article • DOI:10.1103/PHYSREVE.96.023307

Versatile and efficient pore network extraction method using marker-based watershed segmentation. — [Source link](#)

Jeff T. Gostick

Institutions: University of Waterloo

Published on: 16 Aug 2017 - Physical Review E (American Physical Society)

Related papers:

- [Pore-network extraction from micro-computerized-tomography images](#)
- [Pore-scale imaging and modelling](#)
- [OpenPNM: A Pore Network Modeling Package](#)
- [An automated simple algorithm for realistic pore network extraction from micro-tomography images](#)
- [Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/versatile-and-efficient-pore-network-extraction-method-using-1sg2o4ihmo>

Versatile and efficient pore network extraction method using marker-based watershed segmentation

Jeff T. Gostick*

University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

(Received 21 April 2017; published 16 August 2017)

Obtaining structural information from tomographic images of porous materials is a critical component of porous media research. Extracting pore networks is particularly valuable since it enables pore network modeling simulations which can be useful for a host of tasks from predicting transport properties to simulating performance of entire devices. This work reports an efficient algorithm for extracting networks using only standard image analysis techniques. The algorithm was applied to several standard porous materials ranging from sandstone to fibrous mats, and in all cases agreed very well with established or known values for pore and throat sizes, capillary pressure curves, and permeability. In the case of sandstone, the present algorithm was compared to the network obtained using the current state-of-the-art algorithm, and very good agreement was achieved. Most importantly, the network extracted from an image of fibrous media correctly predicted the anisotropic permeability tensor, demonstrating the critical ability to detect key structural features. The highly efficient algorithm allows extraction on fairly large images of 500^3 voxels in just over 200 s. The ability for one algorithm to match materials as varied as sandstone with 20% porosity and fibrous media with 75% porosity is a significant advancement. The source code for this algorithm is provided.

DOI: [10.1103/PhysRevE.96.023307](https://doi.org/10.1103/PhysRevE.96.023307)**I. INTRODUCTION**

Porous materials play a vital role in many technologies and industries. Naturally occurring porous media such as rock and soil have been the subject of intense study in the context of oil recovery and aquifer management [1]. Engineered or manufactured porous materials have enabled breakthroughs in many fields such as electrodes [2–4], membranes [5], biomedical applications [6], and more [7]. Designing porous materials represents a classic optimization problem: The presence of the solid is necessary to provide some key function such as reactive surface area, but it also creates obstacles to flow and transport. A defining feature of all porous materials is that the pore structure can have an overriding impact on the transport processes. Two materials with the same porosity will have dramatically different transport properties depending on how the pores are spatially distributed, how they are connected, their shapes, size distribution, and so on. Because of this strong dependence on pore structure, visualization is a major component of porous media analysis. High-resolution x-ray tomography is particularly useful as it allows views of the internal structure [8]. Benchtop x-ray tomography scanners are now widely available that can produce images with $1\text{-}\mu\text{m}$ voxel resolution, and the latest generation can obtain images with 100-nm resolution or lower [9,10]. Extracting the vast amount of information found in such images is an active pursuit [11–16].

In conjunction with this explosion in imaging capacity has been the relentless growth of computational power. It is now technically possible to use these images as a computational mesh and perform simulations of flow or diffusion directly inside the pore structure, often referred to as direct numerical simulation (DNS) [17,18]. Although this approach is feasible, it requires enormous computational resources to simulate domains of any significant size. An alternative to DNS is pore network modeling (PNM), which offers several important

advantages, albeit by making assumptions and simplifications. The main advantage of PNMs is that they can accomplish simulations on many millions of pores rather than hundreds of pores currently manageable with DNS. A second advantage of PNMs is that multiphase flow can be simulated with ease. Percolation theory describes the sequence and pattern of an invading phase, which can be incorporated into the network transport calculations. Taken together, this means that PNMs can be used to study complex multiphase flow problems in large domains, such as entire catalyst pellets [19], electrodes [20], and paper [21]. Developing faithful pore network models of materials is thus essential for the study and advancement of many technologies.

One of the most important structural metrics of a porous material is the connectivity of the pore space, or the so-called pore network. This has bearing on the diffusive tortuosity, percolation of immiscible fluids, relative permeability, and so on. To this end, researchers are highly motivated to extract pore networks from tomographic images. This task is substantially more complicated than obtaining other metrics such as the chord length distribution or the two-point correlation function, which are straightforward applications of basic image analysis tools [22]. As made abundantly clear by Bhattad *et al.* [23] in their excellent review article comparing network extraction techniques, the main challenge is the unclear definition of a pore and throat, and identifying where one pore ends and another begins, especially in three dimensions (3D). The currently favored method for extracting pore networks from images is based on the maximal ball algorithm proposed by Silin and Patzek [24], and refined by Blunt and co-workers [25,26]. Another widely used approach is based on finding the branch points of the medial axis of the pore space [27–31]. Rounding out the methods is the watershed segmentation of the pore space, that was originally investigated by Thompson *et al.* [32] and Sheppard *et al.* [33] over a decade ago, but has received surprisingly little attention until recently [34,35]. The common thread running through the existing network extraction algorithms is that they are essentially

*jgostick@uwaterloo.ca

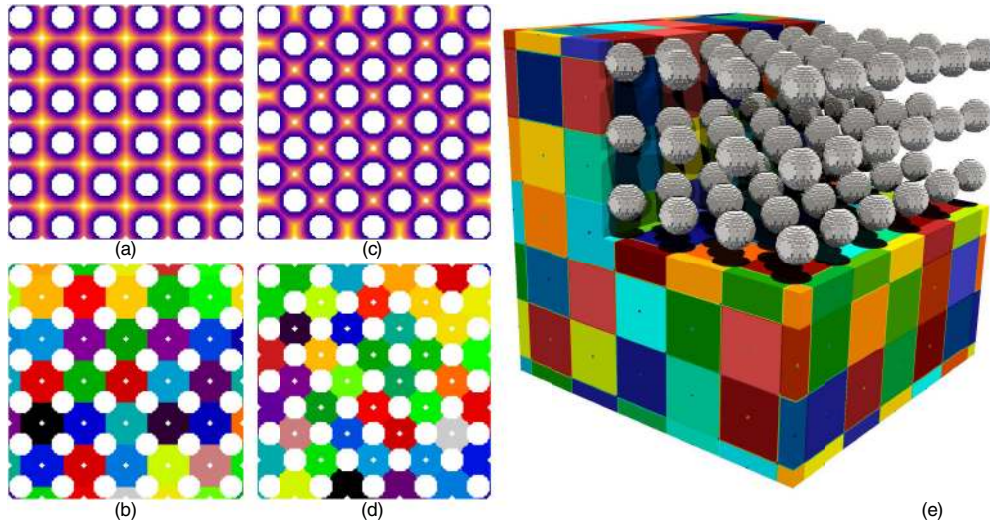


FIG. 1. Illustration of marker-based watershed segmentation procedure on simple images of cubic packings of spheres. (a),(c) Distance transform of void space. (b),(d) Segmented pore space resulting from the marker-based watershed segmentation. (e) Segmentation results on 3D body-centered-cubic packing showing a slice of the sphere locations.

optimized for rock and soil, due to the importance of these media in oil recovery and aquifer management. Other porous materials, such as paper and felt, bone and biomaterials, foam, filters, and so on have been less studied. An example of this geology-centric view is the recent work by Yang *et al.* [36] who developed a unique network extraction algorithm based on progressively dilating the solid phase and monitoring the void space for the appearance of isolated regions, which corresponds to pores being pinched off from the network. Despite its ingenious simplicity, this method is unsuitable for highly porous materials where the pinching of the solid phase would require far too much dilation.

In this work, an algorithm is presented that can extract the pore network from high-porosity materials based on the watershed segmentation approach mentioned above. It is an improvement on the recently reported algorithm of Rabbani *et al.* [34], whose focus was on sandstones; it was not directly applicable to high-porosity media. The main issue is that for high-porosity materials, the distance transform contains not only peaks at each pore center, but also many ridges and plateaus that appear as local maxima, which leads to a highly oversegmented image. Rabbani *et al.* [34] did not completely account for these artifacts, so the present algorithm includes steps to manage the spurious local maxima that give rise to the oversegmentation. The result is an algorithm that actually applies to both classes of materials. The algorithm begins by finding all the peaks that would lead to an oversegmented watershed, then progressively reducing the number of local maxima by ruling out several classes of peaks, producing a subnetwork of the oversegmented watershed (SNOW). Agaesse *et al.* [35] have presented a rough outline of a similar extraction algorithm applied to fibrous media, but they only explored two alternative methods for removing spurious peaks and the results were not scrutinized. The SNOW algorithm, meanwhile, uses different methods, and combines several of them to arrive at a more complete solution that is validated against several well-studied materials. The entire algorithm is less than 60 lines long and is developed entirely in PYTHON

using only open-source and freely available tools. The source code is provided in the Supplemental Material [37] of this work. It takes just a few minutes to process a 500^3 image on a normal personal computer.

II. ALGORITHM

Watershed segmentation has tremendous potential for extracting pore networks from tomography images, since one could argue that the definition of a pore basically corresponds to a catchment basin on a contour map [38,39]. Figure 1 illustrates the process on several basic, “well-behaved” structures. The interstitial space between four disks represents a pore, and the constriction between two disks represents a throat connecting two pores. The color scale in Figs. 1(a) and 1(c) is the Euclidean distance (transform or distance map of the void space, where brighter spots correspond to regions further from the solid). Local brightness peaks occur at pore centers as marked with small white dots. These peaks are found by applying a maximum filter with a spherical structuring element of radius R , then finding where the values in the filtered image are equal to the distance map. The peaks are then passed as “markers” to a marker-based watershed algorithm that finds the basins of the distance map, yielding the segmentation of each pore region as shown in Figs. 1(b) and 1(d). For this idealized geometry, even for the 3D image shown in Fig. 1(e), the process is trivial since the pore bodies and throat constrictions are so uniform and well defined.

In more realistic, random media things become challenging. Figure 2 shows a two-dimensional (2D) image of randomly placed overlapping disks with a relatively high porosity of 60%. The image is 400×400 and the disks are 10 pixels in radius. Though simplistic, this geometry is sufficient to illustrate the problems with the basic watershed approach that are addressed by the present algorithm. When performing the steps outlined above, the results are not acceptable. Although peaks have been found at all the correct locations (i.e., visually identifiable pore centers such as the peaks marked A), the

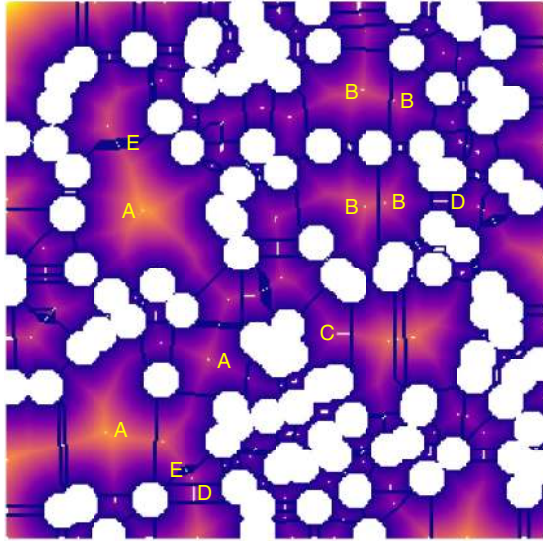


FIG. 2. Illustration of marker-based watershed segmentation procedure on a simple 2D image of overlapping disks. Peaks in the distance transform are identified as locations where a maximum filter with a structuring element of radius $R > 1$ is equal to the distance transform. Without intervention it is clear that the image is oversegmented due to the identification of many spurious peaks that are not true local maxima resulting in many incorrect basins.

image also includes erroneous peaks, generally falling along the ridges in the distance transform. This problem occurs because it is difficult to distinguish between a true peak and a ridge or saddle point, which appear locally maximal since none of the nearby neighboring points are higher. Elimination of the spurious peaks is the main problem that must be surmounted to achieve segmentation of such images into pores. Figure 2 includes labels pointing to the various types of erroneous peaks. The peak marked C lies on a plateau of constant values, despite having higher distance values located to its right, which were out of the range of the structuring element used in the maximum filter. Peaks marked D lie on saddles in the distance map, with higher values on both sides for the same reason. The peaks marked with E are also problematic, and lie on plateaus or saddles, but they arise from the limited resolution of the image. Finally, occasionally two legitimate peaks lie very near each other, such as those labeled B, which should rightfully be considered a single large pore. The algorithm outlined in the present work produces a subset of the oversegmented watershed that results from direct application as outlined above, by removing spurious peaks using a set of common and custom image analysis routines.

A. Detailed description of the SNOW algorithm

The SNOW algorithm proceeds in four main steps: prefiltering the distance map, eliminating peaks on saddles and plateaus, merging peaks that are too near each other, then assigning void voxels to the appropriate pore using a marker-based watershed. These are explained in detail in the following subsections. The code is provided as SNOW.py in the Supplemental Material [37] and is implemented in PYTHON relying heavily on the SCIPY [40] stack.

1. Preparing image

The first step is to obtain the *distance transform* or *distance map* of the pore space [41]. Many software packages offer this function, including MATLAB's IMAGE PROCESSING TOOLBOX, SCIPY's NDIMAGE module [42], and IMAGEJ [43]. The voxelated nature of the binary image creates some artifacts in the distance map. The main problem is that solid walls that are in reality curved appear instead as flat surfaces. When this happens in throat constrictions, the solid faces running parallel to each other create plateaus in the distance map that are misidentified as peaks. Although the objective of the present algorithm is to remove spurious peaks, it is much easier if they are minimized to start with. For this purpose, it is advantageous to run some sort of filter on the distance map to smooth the image and remove or at least minimize the occurrence of such plateaus. Rabbani *et al.* [34] used a median filter for this purpose, but in the present work it was found that a Gaussian blur filter gave much better results since it literally smoothed ridges by incorporating information about the local neighborhood. The median filter is actually recommended when preserving edges is desired, which is not the case here. Agaesse *et al.* [35] applied a gray-scale reconstruction, sometimes referred to as an H-minima filter, that converts peaks to plateaus to remove small local peaks [44]. As discussed below, however, plateaus are also problematic and should be avoided. Figures 3(a) and 3(c) show the peaks obtained before and after applying the Gaussian filter to the distance map, and Figs. 3(b) and 3(d) show the resulting watershed segmentation. The sigma or standard deviation of the convolution kernel is an adjustable parameter, and its impact on the results is explored in Fig. 4 (left). The number of local maxima found in the image (explained in the next section) drops dramatically above sigma of 0.2 indicating this level of smoothing dramatically reduces the spurious peaks. The result is nearly independent of sigma until above 0.4 when the number of peaks starts to decline, indicating that information is being lost by oversmoothing. A sigma value of 0.35 was therefore used throughout this work.

2. Identification of peaks in the distance map

The algorithm proceeds by identifying peaks in the smoothed distance map using a *maximum filter* with a spherical structuring element of radius R . A maximum filter replaces each voxel in the image with the maximum value found within its neighborhood defined by the structuring element. Local peaks retain their value while all other voxels are overwritten with a larger value. Peaks can then be found by locating voxels in the filtered image that are equal to the distance map, resulting in a Boolean mask containing true values at the peaks. It is at this stage of the processing where the difficulties arise, as it is not trivial to find actual local maxima. When R is small, many spurious maxima are found, typically on plateaus and ridges because the neighborhood does not extend beyond the plateau. If R is too large, then some maxima are missed, particularly in pores that are much smaller than the structuring element. Figure 4 (right) shows the number of peaks identified in the image as a function of the radius of the structuring element used in the maximum filter. There is a precipitous drop below $R = 5$, but there is no clear plateau. To ensure no information is lost, an R of 5 provides a good balance of finding all true

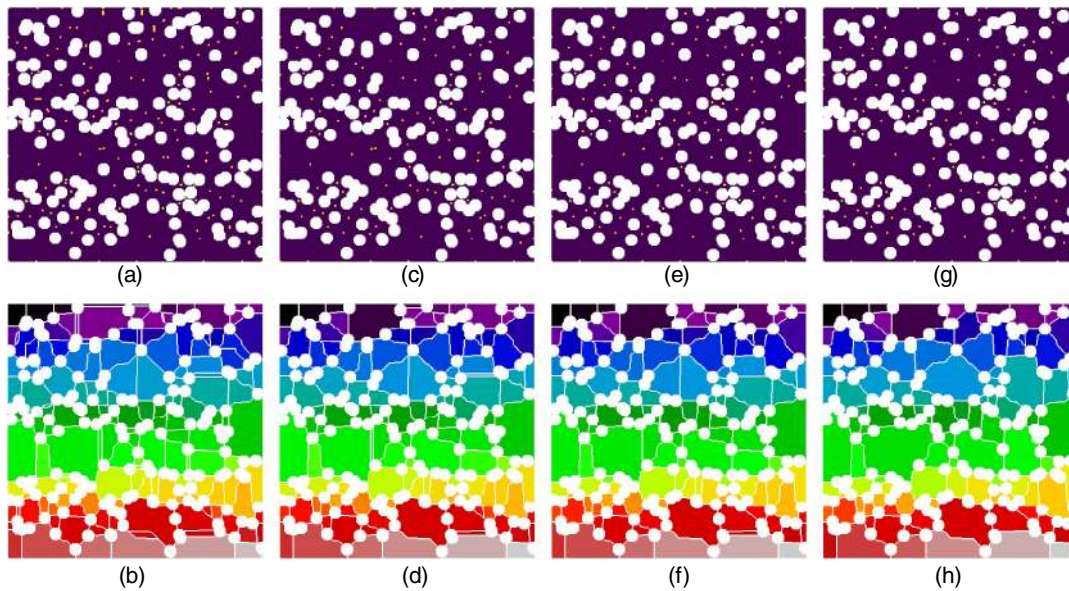


FIG. 3. Illustration of the SNOW algorithm to remove spurious peaks. The top row shows the peaks of the distance transform at each step, and the bottom row shows the resulting segmentation of the pore space. The progressive elimination of spurious peaks can be seen from left to right. The Gaussian blur of the distance transform produces the most striking improvement [column 2 (c),(d)]. The elimination of saddle points removes thin regions between pores [column 3 (e),(f)]. Merging peaks that are near each other prevent the bisection of large continuous regions [column 4 (g),(h)].

maxima without too many spurious peaks. Also shown in Fig. 4 (right) is the impact of prefiltering the distance map with different levels of Gaussian filter. With no filter, there are always extra peaks, while with sigma of 0.25 or 0.5 the results are essentially identical. Note that the suggested values of $R = 5$ and sigma = 0.35 in the Gaussian blur might not translate to images with lower resolution or smaller pores, but this is not explored further here.

3. Removing peaks on saddles

Once a set of peaks has been found, the next step is eliminating the spurious peaks that lie on saddles and plateaus

of the distance map. These peaks were erroneously identified in the previous step because they are surrounded by several voxels of the same distance value, but are ultimately connected to voxels with higher values as the ridge extends out into the open pore space; this is illustrated in Figs. 5(a) and 5(b). A custom procedure was developed to identify these peaks as shown in Figs. 5(c)–5(e). Each peak is analyzed individually using the following iterative procedure: In step (i), the peak is dilated with minimal cubic structuring element ($3^{N-\text{dim}}$) voxels. This dilated peak is then flooded by the maximum value in the underlying distance map as shown in step (ii). Lastly, the flooded dilation is compared to the distance map, and all voxels that are equal in these two images are considered

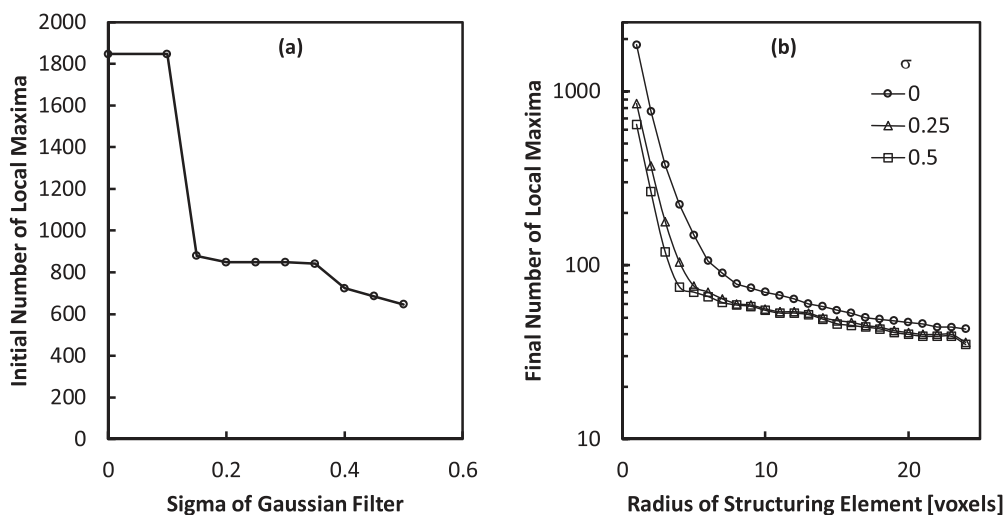


FIG. 4. (Left) Impact of Gaussian filter parameter sigma on the number of local maxima in the image. (Right) Impact of structuring element size on number of final peaks found after applying a maximum filter.

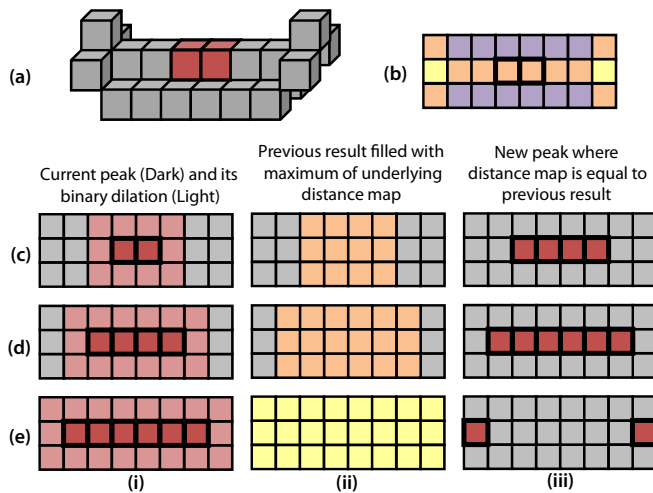


FIG. 5. Schematic diagram showing saddle finding method. (a) Shows a 3D view of the saddle contour, (b) shows a distance map, and (c-i)–(c-iii) show the progression of steps that determine if the peak lies on a saddle.

the new peak(s) (iii). Before repeating the above steps, the new peaks are compared to the old peaks. If the old peaks are a subset of the new peak(s), as shown in Figs. 5(c-iii) and 5(d-iii), then steps (i)–(iii) are repeated. If they are the same, then this peak is a true local maximum and the search stops. If the old peaks are *not* a subset of the new peaks, as shown in Fig. 5(e-iii), then a saddle point was found, also ending the search. These peaks are then removed from the original peaks image as shown in Fig. 3(e). Removal of these peaks generally leads to the elimination of thin regions spanning between solids as shown in Fig. 3(f).

4. Merging nearby peaks

In high-porosity materials, the pores are usually large compared to the solid features, and these large pore spaces can lead to peaks that are quite near each other as depicted by the peaks labeled B in Fig. 2. Spheres centered on these nearby peaks would overlap each other significantly, so generally only one of these peaks should be kept.

Algorithmically, these peaks are identified by finding (a) the distance between each pair of peaks, and (b) the distance of each peak from the solid. When pairs of peaks are found that are closer to each other than the solid, the one furthest from the solid is kept. Finding the distance of each peak to the solid is a simple matter of checking the value of the distance transform at its location. Finding the distance between markers is done by computing a *distance matrix* between peaks. If an image has N peaks, then the distance matrix is an $N \times N$ matrix with the distance between peaks i and j stored in element (i, j) . Thus, for peak i , it is possible to identify peaks that meet the criteria by searching row i for distance values lower than the distance map value for peak i . Figures 3(g) and 3(h) show the result of removing these peaks. Two key results occurred from this step. Firstly, as anticipated, large pores that were previously bisected are now treated as single large pores. This behavior is important since it will alter the size distribution of the network to make it more representative of the real material. Despite Bhatta *et al.* [23] showing that this has surprisingly little impact on

the transport properties of the network, researchers will expect physically representative size distributions for comparison to other measurements. The second benefit of this step is that several thin regions that were not caught by the saddle point check are removed.

5. Segmenting the image into pore regions

The above steps create a set of markers, which must then be passed to a marker-based watershed algorithm to obtain a segmented image similar to Fig. 3(f). There are several implementations of the watershed algorithm, and not all accept markers as input. The version used here is available in the SCIKIT-IMAGE package (aka SKIMAGE) and is a marker-based implementation [45]. There are a few other technical pitfalls that must be avoided, such as not deleting markers on the edge of the image. The code for the algorithm is provided as SNOW.py in the Supplemental Material [37] so such details can be seen.

B. Obtaining pore network information

The watershed segmentation produces an image with integer values indicating which pore regions (drainage basin) each belongs to, so some additional effort is required to convert this information into an actual pore network model. The following sections describe the type of information that is extracted, and briefly outlines how it is achieved. GETNET.py in the Supplemental Material [37] is PYTHON code that performs these steps and outputs the data in a format suitable for importing into OPENPNM [46]. It should be stressed that the following interpretation of pore and throat size information represents only the most general approach, and much more sophisticated analyses could be brought to bear [47,48].

1. Connectivity

Determining which pores are connected to each other is the foundation of creating a pore network. The watershed segmentation makes this trivially easy. Each region in the watershed corresponds to a pore and each region is adjacent to its neighboring regions, so finding connectivity is just a matter of scanning which labels are found in the layer or shell of voxels adjacent to each region. This can be accomplished by isolating each region, dilating it, and then inspecting which labels in the watershed image are overlapped, as shown in Fig. 6(a). In this figure, the dilation of pore 7 overlaps voxels labeled 11, 64, and 67; hence these three pores are its neighbors.

2. Pore and throat geometric properties

Table I gives a brief description of how each pore and throat property was found. To determine the size, volume, location, *etc.*, of pore i is only a matter of isolating the region of the watershed image with voxels labeled i , and analyzing the region's properties, as illustrated in Figs. 6(b)–6(d). In general, the needed size information is obtained by looking at the corresponding set of voxels in the distance map.

The method used to scan for neighboring pores can also be used to determine throat properties. The overlap of the dilated pore region with its neighbors cuts through the cross section

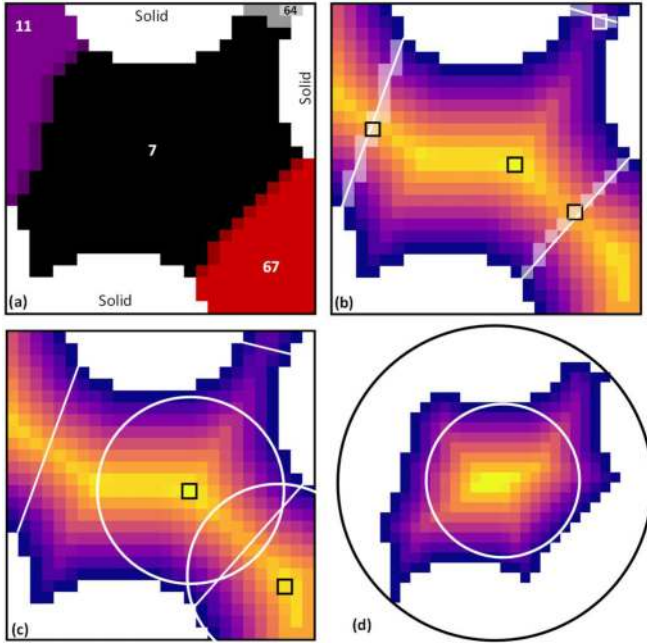


FIG. 6. Illustration of connectivity and size determination for a pore in 2D. (a) The labeled pore regions with pore 6 as the pore of interest, and neighboring pores 14, 28, and 66. The throat regions are found by dilating region 6 and identifying overlapping regions, shown by pale-colored pixels. (b) The distance transform of the image is used to determine the pore and throat size information from the peak values indicated by black boxes. (c) Using the global distance map results in the pore diameter extending into the neighboring pore. (d) Using the distance map obtained within the pore only results in the inscribed pore diameter and a more geometrically consistent pore centroid.

of the throat connecting these two pores. The key difference between the pore and throat calculations is that the throats are 2D surfaces embedded in 3D space, so some of the definitions must be reduced by one dimension (i.e., surface area becomes perimeter).

The actual details of the steps for extracting geometrical information can be altered for different media as needed. GETNET.py in the Supplemental Material [37] is PYTHON code for accomplishing these steps given a distance transform of the pore space and the watershed segmentation produced by the SNOW algorithm. The validity of the extracted pore network is analyzed in detail in Sec. III.

C. Performance

X-ray tomography devices routinely produce images of 1000^3 voxels or more, so any algorithm must be able to analyze such massive datasets in a reasonable time. It is also desirable that standard computational resources such as desktop workstations can accomplish the job, rather than relying on supercomputers. Ensuring good performance was an important criterion when developing the SNOW algorithm. At its core, SNOW relies heavily on standard image analysis tools, specifically the distance transform, maximum filter, Gaussian blur, and the marker-based watershed segmentation. These are available in any environment that offers comprehensive

image analysis tools, including IMAGEJ, PYTHON (via SCIPY and SKIMAGE), and MATLAB. In the present work, the algorithm is implemented in PYTHON using NUMPY to accelerate array operations by vectorization, as well as SCIPY's NDIMAGE module and the SKIMAGE library, which are both built on NUMPY. Figure 7 shows the run times for the key image analysis routines as a function of image size, ranging from 100^3 to 800^3 . All the functions scale approximately linearly. The marker-based watershed is clearly the slowest step, but the distance transform and maximum filter require non-negligible times. At present, these libraries are not universally parallelized. It could be well worth the effort to implement SNOW in a programming language that offers parallelized implementations. For instance, the distance transform function in IMAGEJ is significantly faster than the SCIPY.NDIMAGE version.

Not shown in Fig. 7 is the time required to analyze the segmented image to extract the pore and throat information, which takes about half as long as the watershed step. This function, as implemented in GETNET.py in the Supplemental Material [37], scans each pore region in isolation to extract size information. This for-loop could easily be done in a parallel manner by assigning a fraction of the pores to each available core, using PARFOR in MATLAB or JOBLIB in PYTHON, for instance. Despite the possible speed improvements, the overall time required for the SNOW algorithm is quite decent. On a laptop with an i7-6700HQ processor and 16 GB of RAM, the entire network extraction process on a 400^3 image takes 142 s.

Although it is fairly light on computational resources, the SNOW algorithm can require a machine with a large amount of RAM. This is unavoidable based on the size of the images (i.e., billions of voxels) coupled with the fact that numerous copies of the image must be stored. This problem could be addressed by reducing the precision of the images, but in general, a machine with 16 GB of RAM is sufficient for an image of 400^3 voxels. A 1000^3 image was extracted in less than 1 h, but it required more than 50 GB of RAM.

III. VALIDATION

In this section, the SNOW algorithm is applied to a variety of different images of porous materials, and the results are compared to known accepted values. Capillary pressure curves and permeability calculations were performed using OPENPNM, an open-source pore network modeling tool [46].

A. Voronoi cells

The Voronoi graph has been used by several pore network modelers as a proxy for a random fibrous medium [21,49,50]. With this approach the edges of the Voronoi graph represent fibers, the Voronoi cells define pores, and the facets between cells are the throat constrictions. The Voronoi graph provides an excellent test case for a network extraction algorithm, since (a) once the Voronoi graph is constructed all the topological and geometric information is known, and (b) it is possible to construct a 3D voxel image of the Voronoi edges to run the extraction algorithm on. The properties of the network extracted from the image can then be compared to the properties of the generated Voronoi network, providing a very direct validation. On the downside, despite being random, a

TABLE I. Description of geometric properties for pores and throats extracted from the images.

| Pore properties | Description |
|------------------------|---|
| Volume | The volume of a region can be easily found by summing the number of voxels in the region [i.e., all the black voxels in Figure 6(a)]. This value is not the same as the volume of the pore body that lies in the region, which can be found using the diameter and the appropriate geometric formula. |
| Extended diameter | This value is found as the maximum value of the <i>global</i> distance map lying within each pore region. As shown in Fig. 6(c) this definition means that the pore diameter can extend into neighboring pores. These extended pores tend to overlap each other, which creates inconsistencies such as negative throat lengths and double counting of pore volume. |
| Inscribed diameter | This is obtained in the same manner as the extended diameter, but a <i>local</i> distance map of just the pore region is used. This confines the pore body entirely inside its region as shown in Fig. 6(d). Unless otherwise stated, this value is used as the pore diameter in all calculations. |
| Surface area | Calculation of area is quite difficult in voxel images. It is simple to count the number of voxels on the surface of a volume, but it is not at all simple to know how much area that represents. The number of surface voxels is found as the number of voxels in the <i>local</i> distance transform that have a value of 1. Following Dong and Blunt [25] the area is estimated as the number of voxels on the surface multiplied by the area of one voxel face. |
| Solid area | This is found as the number of voxels in the <i>global</i> distance transform with a value of 1. This means that voxels lying in a throat are not counted since in the global distance map they are far from the solid. |
| Coordinates | The pore coordinates can be taken as the location of the peak in the global or local distance map, or the centroid of the region. The latter was used here since finding peaks in the distance map is a complex issue, as outlined above. This choice means that the pore diameter value does not necessarily correspond with the pore center location, but the shift is small. |
| Throat properties | Description |
| Inscribed diameter | As shown in Fig. 6(b) it is possible to identify the diameter of the throat from the maxima of the global distance transform. |
| Circumscribed diameter | This value is not of direct use in calculations, but it could be used to gauge the noncircularity of a throat. It is found by calculating a distance matrix between all voxels in the throat, then finding the largest value in this matrix, which gives the distance between the two most separated voxels. |
| Equivalent diameter | This value is defined as the diameter of a circle with the same area as the throat cross section. Finding the throat cross-sectional area faces the same challenges as for the pore surface area described above, namely, how to convert the number of voxels into an area. The same approach is taken here, by counting the number of voxels in the throat and multiplying by the area of one voxel face. |
| Perimeter | This is found by counting the number of voxels in the throat with a distance transform value of 1, then multiplying this by the voxel length. |
| Centroid | This is found as the location of the maximum value in the distance map. A center of mass calculation, as done for the pore body, is not reliable here since the throat is a 2D surface and likely curved, so the center of mass could potentially lie outside it. |
| Total length | The total length between two pores is calculated as the Euclidean distance between the two pores' centroids passing through the throat centroid. |
| Direct length | This is calculated in the same manner of the total length, but without passing through the throat centroid. This is the approach used by Dong and Blunt [25] so is included here. |
| Length | This is the actual throat length used in transport calculations and it is equal to the total length minus the radius of each neighboring pore. |

Voronoi image is not the most challenging to extract since the pore bodies and throats are all clearly defined by the convex hulls of the Voronoi graph.

Figure 8 shows a Voronoi graph that has been converted to a fibrous image by drawing 1-voxel-thick lines between each Voronoi vertex, then dilating them to make thicker fibers (in this case 9 voxels). If the voxels are taken to be $1 \mu\text{m}^3$ then the fiber diameter is $9 \mu\text{m}$, the domain is $0.20 \times 0.40 \times 0.40 \text{mm}^3$, and the porosity is 85%. Overlaid in this figure is the network as extracted by the SNOW algorithm with spheres indicating pores proportional to the inscribed diameter, and

thin white lines for each throat connection. Visual inspection suggests that a decent qualitative match was achieved as the pore network lies approximately where it should, but additional validation is required.

The number of pores (N_P) and throats (N_T) in the generated Voronoi network was $[N_P, N_T] = [500, 2180]$, while the extracted network had $[N_P, N_T] = [493, 2282]$, using a sigma of 0.4 in the Gaussian filter and $R = 5$ in the maximum filter. The missing pores are either due to small regions being neglected or large neighboring regions being combined. The additional throats are probably due to the way connectivity is found by

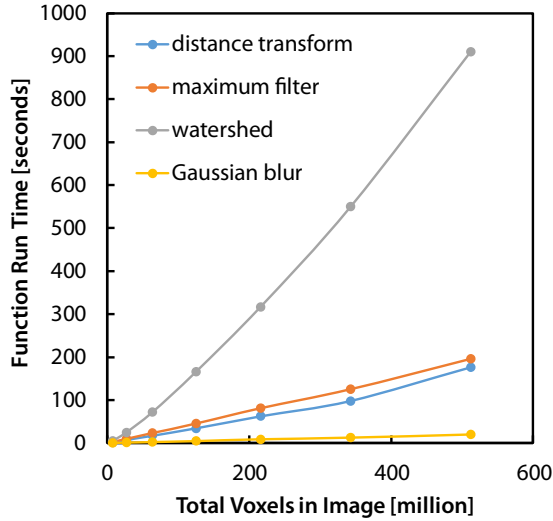


FIG. 7. Performance of the key image analysis routines used in the SNOW algorithm as a function of image size. Tests were performed on a Dell Precision 7550 Workstation with a Xeon E2660 Processor (12 cores) and 72 GB of RAM. The largest size corresponds to an 800^3 image.

dilating the pore region, which may have created overlaps at the Voronoi vertices which are shared by many pores. Though not perfect, the differences represent a small percentage error in the total number of pores and throats.

The validity of the extracted network can be more deeply analyzed by comparing size information. Figure 9 shows the capillary pressure curves obtained from both the generated and extracted network. Additionally, the capillary pressure curve obtained using morphological image opening [49,51] is shown. The agreement between all three is quite good. It has previously been shown that the generated Voronoi agrees well with the morphological image opening result [49] when the throat capillary pressure is calculated using the diameter of

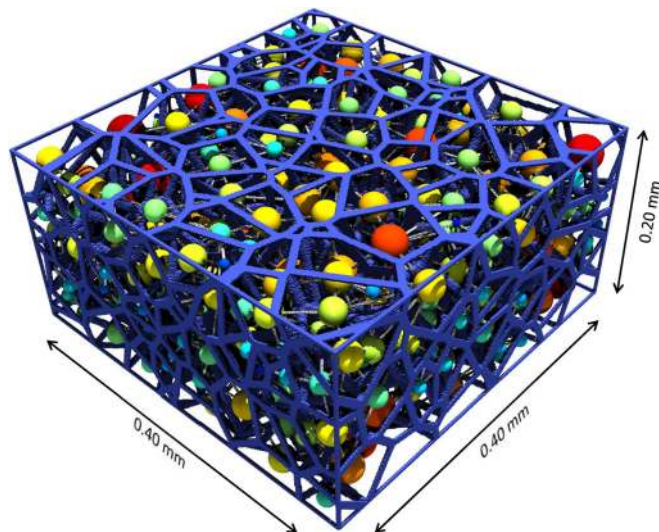


FIG. 8. Generated Voronoi graph to represent fibers (blue voxels) with the extracted network overlaid. The pores are represented by spheres that are smaller than actual size to improve visualization.

an inscribed circle, which was used here as well. The image analysis approach shows a slower approach to full saturation at high capillary pressures as the invading phase inflates to fill corners and small features that are not captured by either pore network. At the low-pressure range, the extracted network rises slightly later and faster, which suggests that the surface pores are not being extracted as faithfully as possible. Overall, the ability of the extracted network to match the generated network so closely is a strong indication that the correct throat sizes and connectivity have been captured. This can be more explicitly confirmed by comparing the size distributions directly, as shown in the inset of Fig. 9. The distributions for pore and throat diameter both agree very well. The pore-to-pore distance as measured through the centroid of the connecting throat is also in close accord. This particular metric indicates that the topology of the extracted network is valid since both the pore and throat centers were found correctly. Finally, the coordination number shows a similar distribution, but with a few higher values for the extracted network. This is clearly related to the fact that additional throats were found in the extracted network, probably due to the dilation method creating additional overlaps at the vertices. Despite some small discrepancies, the geometrical and topological metrics are in excellent agreement.

Finally, the permeability coefficient was calculated for both the networks. Because fibrous media are so open and the pores are essentially open space, the hydraulic conductance was calculated by assuming no pressure drop in the pores. This means that the throat length was equal to the center-to-center distance between pores, less the radius of each neighboring pore, which was found from the size of an inscribed sphere. The hydraulic conductance between pores j and k is thus controlled only by the properties of the throat between them, as

$$g_{H,j-k} = g_{H,t} = \frac{\pi}{128\mu} \left(\frac{D_t^4}{L_t} \right), \quad (1)$$

where D_t is the diameter of the throat, L_t is the length of the throat, and μ is the viscosity of the flowing phase. The flow rate between pores j and k is found from

$$q_{j-k} = \frac{\pi}{128\mu} \left(\frac{D_t^4}{L_t} \right) (P_j - P_k) = g_{H,t} \Delta P_{j-k}, \quad (2)$$

where P is the pressure in the pore and q is the flow rate between them.

This yielded $K = 16.82 \times 10^{-12}$ and $K = 19.26 \times 10^{-12}$, for the generated and extracted networks, respectively (all in units of m^2), when using the inscribed diameter for D_t . The Voronoi network was not generated with any anisotropy so these values were nearly the same in each direction as expected. The impact of anisotropy will be considered in detail in Sec. III C when the SNOW algorithm is applied to images of real fibrous media. The permeability coefficients of the two networks were within 10% of each other, which is more than acceptable. In principle, the actual permeability of the voxel image could be obtained using a direct numerical simulation such as the lattice-Boltzmann method. As a simpler alternative there are many correlations that can predict the permeability of a randomly oriented arrays of fibers [52,53]. Jackson and James [52] proposed the following as a general fit to a large

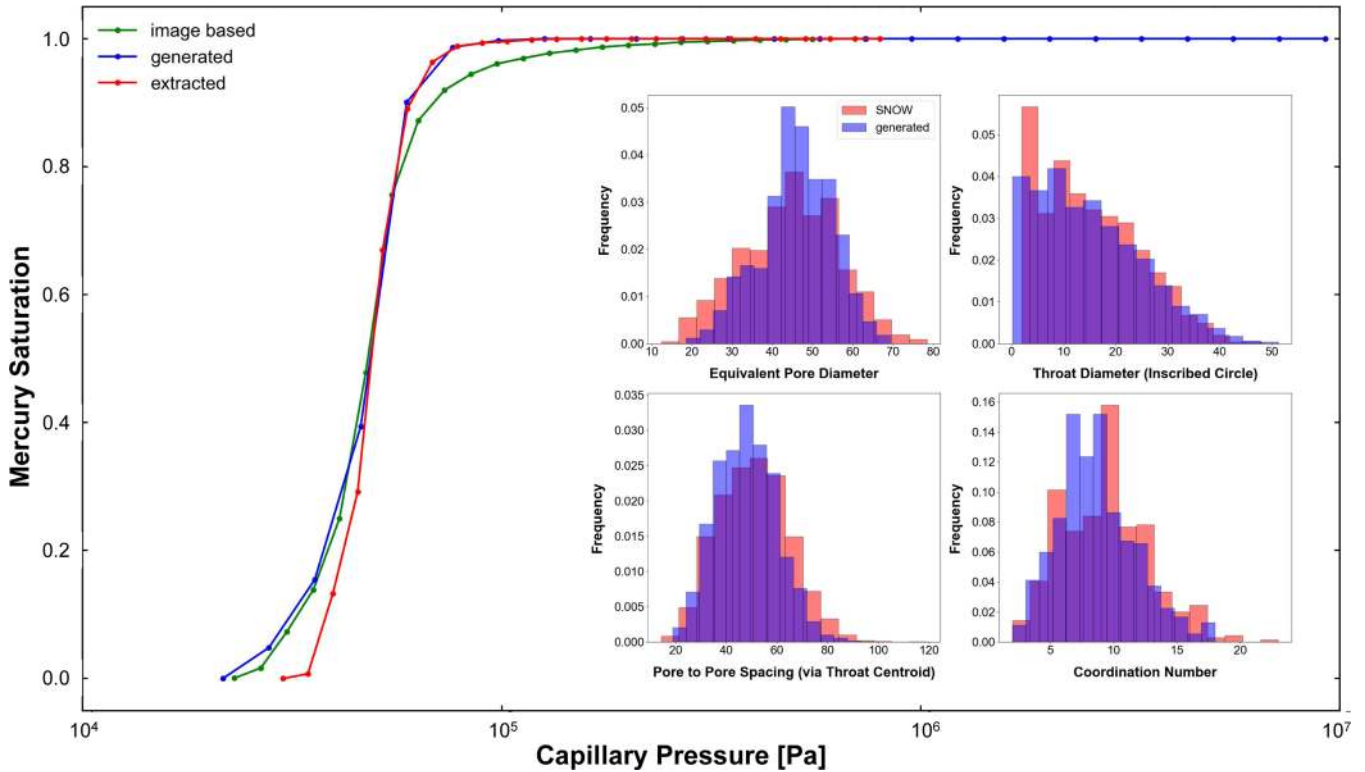


FIG. 9. Comparison of generated Voronoi pore network (blue) with the network extracted from a voxel image of its vertices (red). Mercury intrusion was simulated in both networks as well as using morphological image opening of the Voronoi image (green), yielding very consistent results. Inset: Various size distributions of the generated and extracted networks.

data set:

$$K = \frac{3d_f^2}{80(1 - \epsilon)} [-\ln(1 - \epsilon) - 0.931], \quad (3)$$

where K is the permeability coefficient in m^2 , ϵ is the porosity, and d_f is the fiber diameter in meters. This equation predicts a permeability coefficient of $19.6 \times 10^{-12} m^2$, which is in excellent agreement with both model predictions. This match was achieved without using a shape factor for the pore network conductance values, but the pressure drop in the pore bodies was assumed to be zero.

Given the ease of predicting permeability values using the above correlation, a further check of the SNOW algorithm was performed by dilating the fibers in the Voronoi image with a spherical structuring element of size $R = 3$. This effectively increased the fiber diameter by 6 voxels to $15 \mu m$ and reduced the image porosity to 0.68. Inserting these values in the correlation gives $K = 5.01 \times 10^{-12} m^2$, while reapplying the SNOW algorithm to the altered image produced a network with an average permeability of $K = 5.34 \times 10^{-12} m^2$. Such a close match provides a strong validation of the SNOW algorithm, especially given the subtleness of changing fiber thickness and porosity which does not affect topology.

B. Berea sandstone

As another test case, the SNOW algorithm was applied to Berea sandstone, which is a standard material used in geoscience studies. Pore network extraction algorithms have been applied to Berea since the earliest attempts by Silin and

Patzek [24], Dong and Blunt [25], and others [29,30]. For this validation, the network extracted by the SNOW algorithm was compared to that of Dong and Blunt [25] since they provide their extracted network as well as the tomography image for other users to compare [54]. Moreover, their implementation of the maximal ball algorithm is in wide use so it is important to ensure the SNOW algorithm agrees.

The first comparison is to note that the number of pores and throats in the maximal ball network is $[N_P, N_T] = [6004, 12067]$, while the SNOW extraction has only $[N_P, N_T] = [4181, 8431]$. This substantial difference is analyzed in more detail below. The results of the extraction can be seen in Figs. 10(a) and 10(b), and the pores appear to be located in the correct locations which is encouraging but not conclusive. Figures 10(c) and 10(d) compare the stick-and-ball representation of the maximal ball and the SNOW networks, respectively. Opposite to expectation, the SNOW network appears to have a higher density of pores, but this is an illusion for two reasons. Firstly, the pores in the SNOW network are larger, so their red spheres appear to the eye to be more prevalent. Secondly, the SNOW network has lower pore and throat density, so the provided view is actually deeper into the network revealing more red spheres. Despite the first appearances, these networks are actually quite similar in terms of topology, spatial distribution, throat orientation, spacing, and pore and throat sizes, which again is encouraging but not conclusive.

The inset of Fig. 11 compares geometric properties of the network extracted by the SNOW algorithm to those determined by the maximal ball algorithm. The maximal ball values were

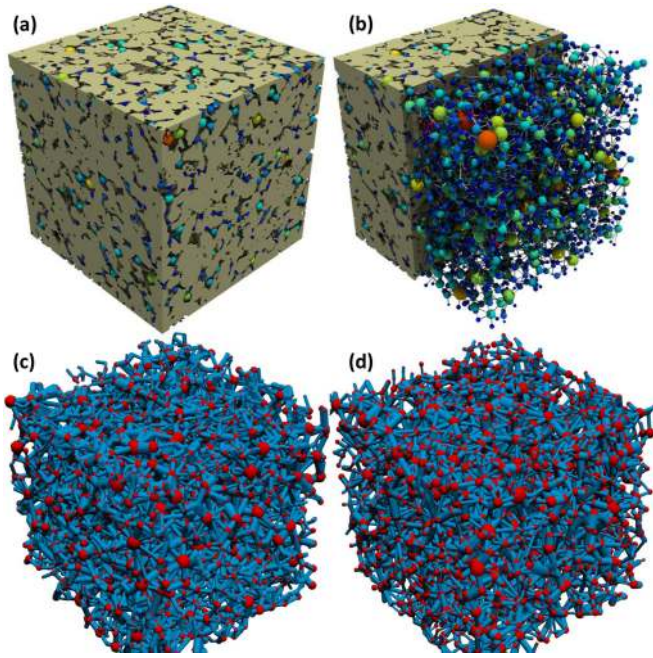


FIG. 10. Views of the Berea sandstone sample. (a) The voxel image showing the solid in gold and the SNOW extracted network overlaid; (b) same as (a) but with some of the solid cut away. (c) The stick and ball representation of the maximal ball extraction, and (d) the stick and ball representation of the SNOW extraction.

obtained by importing the data files provided by the Blunt group into OPENPNM, which has the ability to import the “statoil” format. The agreement between the pore and throat size and the network topology is generally good, but the SNOW extraction has consistently larger pore and throat sizes. This suggests that the SNOW algorithm divides the pore space into larger pieces, which also explains why the SNOW algorithm found fewer pores and throats. The most direct evidence of this is the pore-to-pore spacing, where the distribution for the maximal ball network is uniformly shifted to shorter values indicating that pores are generally closer together. The discrepancy in the number of pores and throats between the two extracted network all stem from the tendency of the SNOW algorithm to identify fewer, larger pores. Determining which division of space is more appropriate or whether it makes any difference can be decided by looking at other network metrics and simulation results.

Figure 11 shows the capillary pressure curves for the maximal ball and SNOW extractions (both using inscribed diameter), along with the morphological image opening result. If the image-based approach is taken as correct, then the SNOW extraction seems to have produced a more faithful representation of the network. The smaller throat sizes of the maximal ball manifest as higher entry pressures that shift the drainage curve to the right. This is consistent with the fact that the maximal ball divides the void space into smaller pores, which naturally have smaller throats between them. The original paper outlining the maximal ball method does not compute capillary pressure curves, so it is difficult to speculate further on this discrepancy. It should be conceded however, that image-based capillary curves are generally shifted to the right compared to experimental data because throat invasion

is controlled by the inscribed diameter, which is smaller than the effective diameter when throats are not symmetrical (i.e., flattened). The capillary pressure curve for the SNOW network matched the image-based curve because it too used the inscribed diameter, but both of these would be higher than actual experimental data. This is explored further when discussing fibrous media in Sec. III C.

Another way to compare the extracted networks is the permeability coefficient, but this is complicated by the various shape factors and segmentation coefficients that were used in the original works. Direct computation using only the pore and throat sizes, assuming cylindrical throats and spherical pores, gives $K = 0.0685 \times 10^{-12} \text{ m}^2$ for the maximal ball network and $K = 0.0801 \times 10^{-12} \text{ m}^2$ for the SNOW network, which is remarkably similar given the differences in the size and density of the pores and throats. This surprising behavior was noted by Bhattad *et al.* [23] and attributed to the increase in the number of flow conduits offsetting the smaller pores. Both of these values are much too low: Dong and Blunt give the experimental value for Berea as $K = 0.650 \times 10^{-12}$, which is nearly $10\times$ higher than the model prediction, and the permeability of the image was calculated using the lattice-Boltzmann method as $K = 1.286 \times 10^{-12}$, yet another factor of 2 higher. Dong and Blunt [25] attribute the differences to the throats not being cylindrical and invoke a shape factor to account for the deviations. Instead of using the shape factor, which they define using terms specific to the maximal ball extraction, it is also possible to use the more general equivalent diameter described in Sec. II B. The equivalent diameter is larger than the inscribed diameter so it accounts for the additional area for flow. Using this value to calculate the hydraulic conductance, and assuming no pressure loss in the pore bodies, the SNOW network permeability rose to $K = 1.13 \times 10^{-12} \text{ m}^2$, which is acceptably close to the lattice-Boltzmann value for the image, and is almost identical to the permeability reported by Dong and Blunt [25] using maximal ball with shape factors.

C. Fibrous media

Extracting pore networks from fibrous media was the ultimate motivation for this work. The high porosity, lack of clearly defined pores and throats, and high coordination numbers (>10) present a significant challenge to existing extraction algorithms that have been optimized for sandstone. Moreover, the anisotropy of fibrous materials adds an additional level of complexity since the length and orientation of voids must somehow be reflected in the extracted network. The SNOW algorithm does not explicitly analyze pore shapes and orientations, but this information is incorporated into the network structure via the throat length calculation. As mentioned in Sec. II B, the throat length is found as the distance between two pore centers, passing through the throat centroid, minus the radius of each pore. Thus a pore region that is elongated in the X direction will have longer X -direction throats compared to other directions since the pore is treated as a sphere. Depending on the morphology and topology of the network, this can create anisotropy in different ways. If the throats are large, they will act as straight, low-tortuosity pathways through the network, while small throats would represent obstructions to flow. In addition, if most of the throats

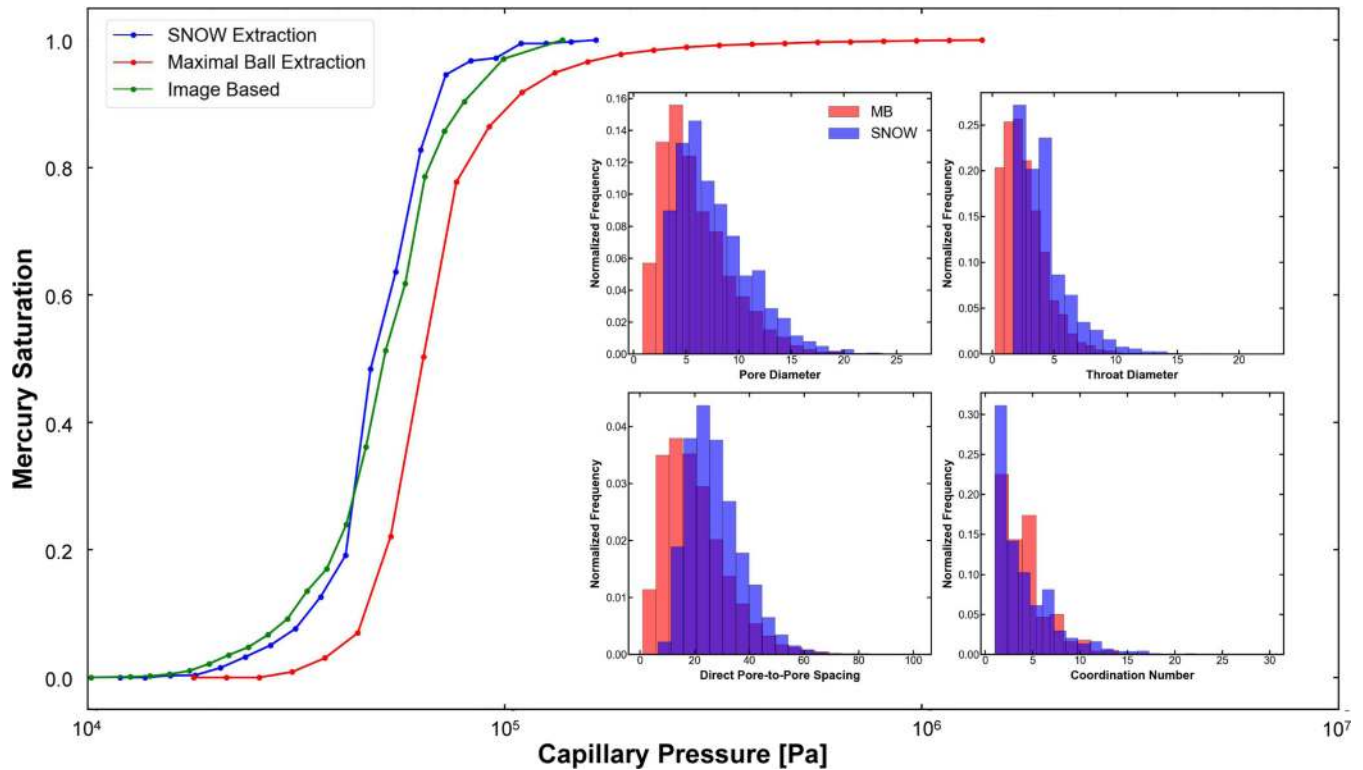


FIG. 11. Comparison of Berea network extracted by the SNOW algorithm (blue) to the maximal ball extraction (red). Mercury intrusion was simulated in both networks as well as using morphological image opening of the Berea image (green), yielding very consistent results. Inset: Various size distributions of the extracted networks.

are oriented in the X direction then flow in other directions could become quite tortuous.

The SNOW algorithm was applied to an image of Toray 120 with no PTFE, used in previous studies [55,56]. This material has a porosity of 75% [57], a thickness of 360 microns, fiber diameter around 10 microns, and a permeability coefficient tensor [58] of approximately $[K_X, K_Y, K_Z] = [15 \times 10^{-12}, 15 \times 10^{-12}, 9 \times 10^{-12}]$, where Z is the through-plane direction (i.e., perpendicular to the fibers). The image itself measured $1500 \times 1500 \times 221$ voxels with a resolution of 1.33 microns. The surfaces of the image have been trimmed to provide a clear and flat boundary for the extraction. About 30 microns were trimmed from each face, which amounts to about—one to two layers of pores. The surface regions tend to have higher porosity than the internal core, so their removal reduced the porosity to 67%. Figure 12(a) shows the voxel image overlaid with the SNOW network. The image was trimmed to $600 \times 600 \times 221$ to improve this visualization. The locations of the pore network generally align with the image, with the largest pores (red) lying in the largest openings, and the smallest pores (cyan) can also be seen filling the smaller gaps. The most striking views are given in Figs. 12(b) and 12(c) where the high degree of throat alignment in the in-plane direction is clearly visible. This feature was expected due to the strong anisotropy exhibited by the fibrous media.

Figure 13 shows capillary pressure curves obtained using several different methods. The image-based analysis and the SNOW extraction using the inscribed throat diameter are in close agreement as has been the case for the materials studied above. This is because throat invasions in the image-

based approach are controlled by the inscribed diameter, so it is a further validation of the SNOW algorithm that a drainage simulation using the inscribed diameter matches. Neither of these curves agree with the experimental mercury intrusion porosimetry (MIP) data, however. In MIP, invasion is controlled by an effective throat shape. Consider the definition of curvature for a simple ellipsoidal curved surface:

$$\frac{P_c}{\sigma} = 2H = \frac{1}{R_1} + \frac{1}{R_2}, \quad (4)$$

where R_1 and R_2 are the radii in perpendicular directions. In symmetrical throats $R_1 = R_2$, but for flattened throats R_2 would be larger, thereby lowering H and P_c . This problem can be remedied by using an alternative means of calculating

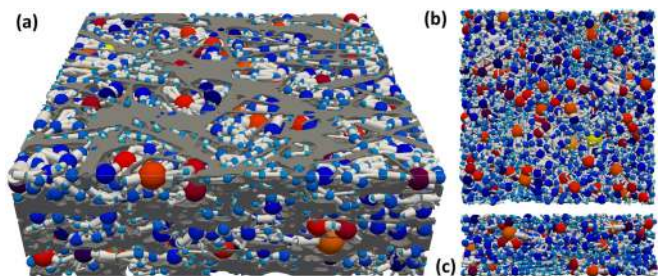


FIG. 12. Views of the fibrous sample Toray 120A (no PTFE) trimmed to $600 \times 600 \times 221$ for easier visualization. (a) SNOW extracted network overlaid with the fiber image (dark gray); (b),(c) are top and side views of the SNOW extracted network, showing the notable amount of in-plane throat alignment.

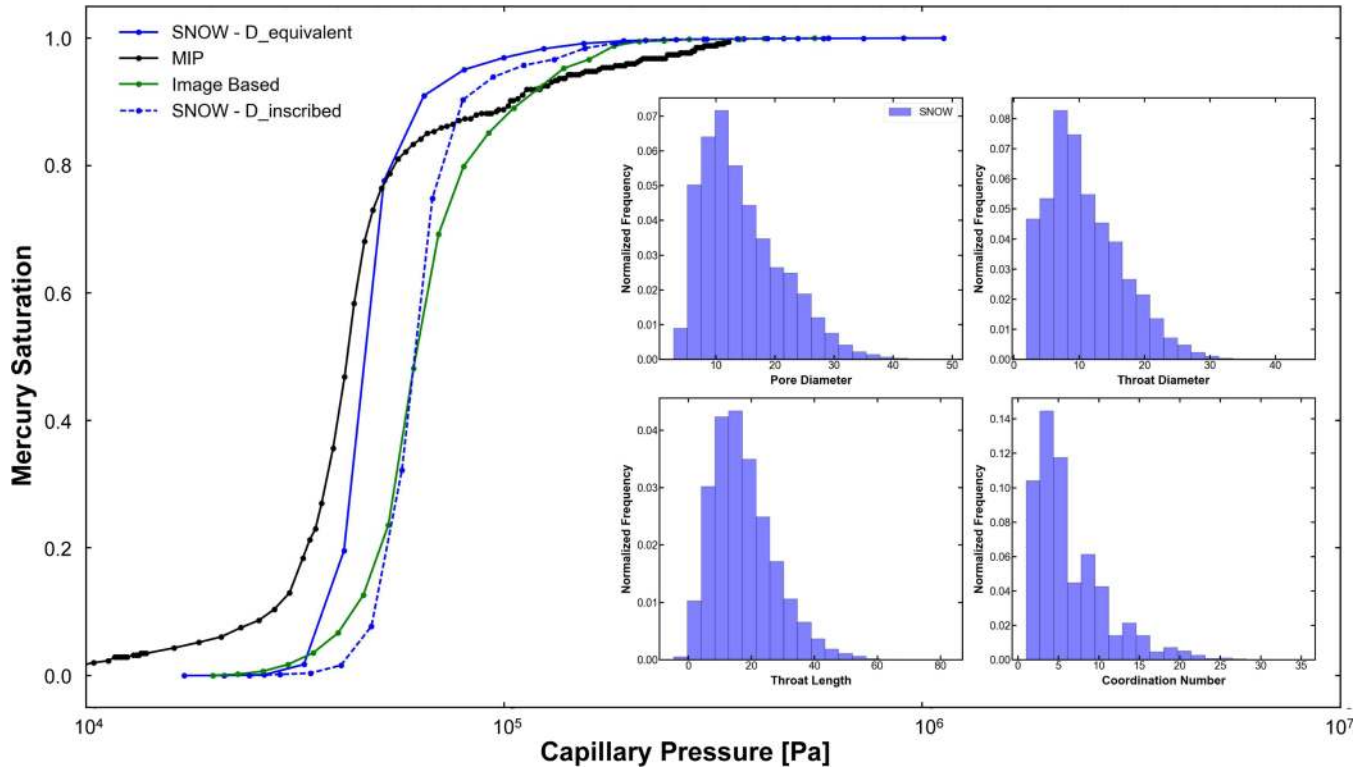


FIG. 13. Capillary pressure curves for Toray 120A (no PTFE) produced by image analysis (green), mercury porosimetry (black), and the SNOW network using either inscribed or equivalent throat diameter. Inset: Various size distributions of the extracted network.

the throat size, such as the equivalent diameter. For throats this means the diameter of a circle with the same area as the throat cross section. This value is difficult to determine due to the voxelated nature of the image, but can be approximated as discussed in Sec. II B. Figure 13 also shows the results of using the approximated value of effective diameter; the capillary pressure curve shifts left and is in better agreement with the MIP data. The size distributions shown in Fig. 13 are in good agreement with accepted values for similar materials [59–62].

The true test of the extraction is matching the known permeability tensor, which is anisotropic. Applying the standard permeability calculation that was used for the Voronoi network in Sec. III A above yielded $[K_X, K_Y, K_Z] = [3.96 \times 10^{-12}, 4.05 \times 10^{-12}, 2.04 \times 10^{-12}]$. Although these values are about $4\times$ lower than the experimental values, they show the expected anisotropy ratio which is a very encouraging result. As with the Berea comparison above, these low permeability values are due to using the inscribed diameter rather than the equivalent diameter. Repeating the simulation with the equivalent diameter in the hydraulic conductance yielded $[K_X, K_Y, K_Z] = [12.4 \times 10^{-12}, 12.2 \times 10^{-12}, 4.54 \times 10^{-12}]$. The agreement with the experimental values is better, and importantly it still includes the expected anisotropy ratio. Although this agreement is good, these values are still somewhat lower than the experimental values. This is likely because the high-porosity surface regions were trimmed from the image, so the overall porosity was notably lower than the full samples on which the experimental measurements were taken. In other words, the predicted permeability values are even closer to the permeability of the image than this comparison suggests.

It is not clear why the equivalent diameter was appropriate for matching the Toray 120 network with the experimental data, while the inscribed diameter gave a better match for the Voronoi network and the correlation. This is most likely due to the anisotropy of the Toray 120 material leading to throats that are quite flattened and thus poorly represented by the inscribed diameter.

IV. CONCLUSIONS

The main motivation for this work was to develop a robust method for extracting pore networks from 3D voxel images of atypical porous materials such as fibrous media, foams, membranes, and so forth. These materials present new challenges that are not well addressed by existing extraction methods, which generally do not attempt to account for high porosity and minimal throat constrictions.

The present algorithm was built upon the idea of using the watershed segmentation of the distance transform proposed by Thompson *et al.* [32] and Sheppard *et al.* [33], and recently revisited by Rabbani *et al.* [34] These works were developed for sandstones, and are not directly applicable to other materials because they do not account for artifacts that arise in high-porosity images with minimal throat constrictions, namely, the presence of spurious peaks in the distance transform. These lead to a highly oversegmented result with many unrealistically small and misshapen pores. The present algorithm altered their procedure to produce a subnetwork of the oversegmented watershed (SNOW) by removing the problematic spurious peaks. Three simple steps were taken: filtering the distance transform with a Gaussian blur to reduce

TABLE II. Network properties obtained by the SNOW algorithm for each image compared to accepted or known values.

| Value | Voronoi ($d_f = 9$) | Voronoi ($d_f = 15$) | Berea | Toray 120A |
|---|-----------------------------|-----------------------------|-----------------------------|-------------------------------|
| Image properties | | | | |
| Dimensions [voxels] | $400 \times 400 \times 200$ | $400 \times 400 \times 200$ | $400 \times 400 \times 400$ | $1500 \times 1500 \times 221$ |
| Resolution ($\mu\text{m}/\text{voxel}$) | 1 | 1 | 5.345 | 1.33 |
| Porosity | 85.0 | 67.5 | 19.5 | 67.6 |
| Network properties | | | | |
| Generated— N_P, N_T | 500, 2180 | | | |
| SNOW— N_P, N_T | 493, 2282 | 521, 1857 | 4181, 8431 | 25338, 79254 |
| MaxBall— N_P, N_T | | | 6004, 12067 | |
| Permeability [$\times 10^{-12}$] | | | | |
| Experimental | | | 0.650 | [15,15,9] |
| Image analysis | | | 1.29 | |
| Correlation | 19.6 | 5.02 | | |
| Generated | 16.2 | | | |
| MaxBall (inscribed) | | | 0.0685 | |
| MaxBall (+shape factor) | | | 1.11 | |
| SNOW (inscribed) | 19.3 | 5.34 | 0.0801 | [3.71, 3.27, 1.25] |
| SNOW (equivalent) | 40.5 | 12.9 | 1.15 | [11.5, 10.0, 3.22] |

the number of bad peaks, removing peaks that actually fell on plateaus and saddles in the distance transform, and removing peaks that were too near another peak. When applied together these steps removed the vast majority of spurious peaks and led to a very satisfactory partitioning of the image into pore regions using a marker-based watershed algorithm. The present work also outlined how to convert this segmented image into a pore network by extracting the pertinent information such as connectivity, pore sizes, and throat sizes. This process of calculating network properties from the watershed image was by no means the final word on the matter [47,48,63], and many improvements could be made such as finding surface area [64] and perimeter more rigorously.

The networks extracted using the SNOW algorithm were compared to several known materials (see Table II). A Voronoi network was generated using OPENPNM, along with a voxel image of the Voronoi edges. The size distributions and permeability of the extracted network agreed very well with the generated Voronoi network. Berea sandstone was also studied, and compared to the network extracted by Dong and Blunt using the maximal ball algorithm [25]. The size distributions showed some differences, with the maximal ball network possessing a larger number of smaller pores. Nonetheless, the permeability values of the maximal ball and the SNOW networks were in excellent agreement with each other, but were $20\times$ lower than the known value for the image. The maximal ball network includes shape factors which bring the

predicted permeability in line with the known values, but these are quite complicated to determine from the image. The SNOW algorithm was able to match the known values by simply using the equivalent throat diameter and neglecting pressure drop in the pore bodies. Additionally, the SNOW network produced capillary pressure curves that were in better agreement with the morphological image opening result than the maximum ball network, which showed higher entry pressures due to its smaller throats. This suggests that the network identified by the SNOW algorithm was closer to reality. Finally, the SNOW algorithm was applied to a fibrous material typically used in fuel cell electrodes. Using the equivalent diameter and neglecting pressure drop in the pores resulted in permeability that matched experimental values very closely. Importantly, the SNOW network included the anisotropy ratio of the material.

The provided algorithm is simple, using mostly standard image analysis functions with a few additional custom functions to trim extraneous points. The code is quite fast, completing the extraction on a 400^3 image in just 142 s. The processing time scales approximately linearly with the total number of voxels in the image. For very large images (1000^3) the process takes less than 1 h, but does require a substantial amount of RAM (>50 GB). Importantly, the SNOW algorithm was able to extract reliable networks from both low-porosity sandstone and high-porosity fibrous media; thus it represents a versatile tool.

- [1] M. Sahimi, Flow phenomena in rocks: From continuum models to fractals, percolation, cellular automata, and simulated annealing, *Rev. Mod. Phys.* **65**, 1393 (1993).
- [2] N. P. Brandon and D. J. Brett, Engineering porous materials for fuel cell applications, *Philos. Trans. R. Soc., A* **364**, 147 (2006).
- [3] P. R. Shearing, D. J. L. Brett, and N. P. Brandon, Towards intelligent engineering of SOFC electrodes: A review of advanced microstructural characterisation techniques, *Int. Mater. Rev.* **55**, 347 (2010).

- [4] P. Trogadas, V. Ramani, P. Strasser, T. F. Fuller, and M.-O. Coppens, Hierarchically structured nanomaterials for electrochemical energy conversion, *Angew. Chem., Int. Ed.* **55**, 122 (2016).
- [5] E. A. Jackson and M. A. Hillmyer, Nanoporous membranes derived from block copolymers: From drug delivery to water filtration, *ACS Nano* **4**, 3548 (2010).
- [6] J. de Jong, R. G. H. Lammertink, and M. Wessling, Membranes and microfluidics: A review, *Lab Chip* **6**, 1125 (2006).

- [7] O. D. Velev and A. M. Lenhoff, Colloidal crystals as templates for porous materials, *Curr. Opin. Colloid Interface Sci.* **5**, 56 (2000).
- [8] D. Wildenschild and A. P. Sheppard, X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems, *Adv. Water Resour.* **51**, 217 (2013).
- [9] M. Andisheh-Tadbir, F. P. Orfino, and E. Kjeang, Three-dimensional phase segregation of micro-porous layers for fuel cells by nano-scale X-ray computed tomography, *J. Power Sources* **310**, 61 (2016).
- [10] W. K. Epting, J. Gelb, and S. Litster, Resolving the three-dimensional microstructure of polymer electrolyte fuel cell electrodes using nanometer-scale x-ray computed tomography, *Adv. Funct. Mater.* **22**, 555 (2012).
- [11] M. Sabharwal, L. M. Pant, A. Putz, D. Susac, J. Jankovic, and M. Secanell, Analysis of catalyst layer microstructures: From imaging to performance, *Fuel Cells* **16**, 734 (2016).
- [12] H. Andrä, N. Combaret, J. Dvorkin, E. Glatt, J. Han, M. Kabel, Y. Keehm, F. Krzikalla, M. Lee, C. Madonna, M. Marsh, T. Mukerji, E. H. Saenger, R. Sain, N. Saxena, S. Ricker, A. Wiegmann, and X. Zhan, Digital rock physics benchmarks—part II: Computing effective properties, *Comput. Geosci.* **50**, 33 (2013).
- [13] J.-F. Delerue and E. Perrier, DXSoil, a library for 3D image analysis in soil science, *Comput. Geosci.* **28**, 1041 (2002).
- [14] R. A. Ketcham, Computational methods for quantitative analysis of three-dimensional features in geological specimens, *Geosphere* **1**, 32 (2005).
- [15] F. Brun, L. Mancini, P. Kasae, S. Favretto, D. Dreossi, and G. Tromba, Pore3D: A software library for quantitative analysis of porous media, *Nucl. Instrum. Methods Phys. Res., Sect. A* **615**, 326 (2010).
- [16] W. Mickel, S. Munster, L. M. Jawerth, D. A. Vader, D. A. Weitz, A. P. Sheppard, K. Mecke, B. Fabry, and G. E. Schroder-Turk, Robust pore size analysis of filamentous networks from three-dimensional confocal microscopy, *Biophys. J.* **95**, 6072 (2008).
- [17] M. J. Blunt, B. Bijeljic, H. Dong, O. Gharbi, S. Iglauer, P. Mostaghimi, A. Paluszny, and C. Pentland, Pore-scale imaging and modelling, *Adv. Water Resour.* **51**, 197 (2013).
- [18] C. Pan, L.-S. Luo, and C. T. Miller, An evaluation of lattice Boltzmann schemes for porous medium flow simulation, *Comput. Fluids* **35**, 898 (2006).
- [19] Y. Liu, F. Meirer, C. M. Krest, S. Webb, and B. M. Weckhuysen, Relating structure and composition with accessibility of a single catalyst particle using correlative 3-dimensional micro-spectroscopy, *Nat. Commun.* **7**, 12634 (2016).
- [20] M. Aghighi, M. A. Hoeh, W. Lehnert, G. Merle, and J. Gostick, Simulation of a full fuel cell membrane electrode assembly using pore network modeling, *J. Electrochem. Soc.* **163**, F384 (2016).
- [21] K. E. Thompson, Pore-scale modeling of fluid transport in disordered fibrous materials, *AIChE J.* **48**, 1369 (2002).
- [22] S. Torquato, L. Gibiansky, M. Silva, and L. Gibson, Effective mechanical and transport properties of cellular solids, *Int. J. Mech. Sci.* **40**, 71 (1998).
- [23] P. Bhattad, C. S. Willson, and K. E. Thompson, Effect of network structure on characterization and flow modeling using x-ray micro-tomography images of granular and fibrous porous media, *Transp. Porous Media* **90**, 363 (2011).
- [24] D. Silin and T. Patzek, Pore space morphology analysis using maximal inscribed spheres, *Phys. A (Amsterdam, Neth.)* **371**, 336 (2006).
- [25] H. Dong and M. J. Blunt, Pore-network extraction from micro-computerized-tomography images, *Phys. Rev. E* **80**, 036307 (2009).
- [26] A. S. Al-Kharusi and M. J. Blunt, Network extraction from sandstone and carbonate pore space images, *J. Pet. Sci. Eng.* **56**, 219 (2007).
- [27] S. Bakke and P.-E. Øren, 3-D pore-scale modelling of sandstones and flow simulations in the pore networks, *SPE J.* **2**, 136 (1997).
- [28] Z. Liang, M. A. Ioannidis, and I. Chatzis, Geometric and topological analysis of three-dimensional porous media: Pore space partitioning based on morphological skeletonization, *J. Colloid Interface Sci.* **221**, 13 (2000).
- [29] M. Prodanović, W. B. Lindquist, and R. S. Seright, 3D image-based characterization of fluid displacement in a Berea core, *Adv. Water Resour.* **30**, 214 (2007).
- [30] A. P. Sheppard, R. M. Sok, and H. Averdunk, Improved pore network extraction methods, in *Proceedings of the International Symposium of the Society of Core Analysts, Toronto, 21–25 August* (Society of Core Analysts, 2005).
- [31] R. I. Al-Raoush and C. S. Willson, Extraction of physically realistic pore network properties from three-dimensional synchrotron x-ray microtomography images of unconsolidated porous media systems, *J. Hydrol.* **300**, 44 (2005).
- [32] K. E. Thompson, C. S. Willson, C. D. White, S. Nyman, J. Bhattacharya, and A. H. Reed, Application of a new grain-based reconstruction algorithm to microtomography images for quantitative characterization and flow modeling, in *Proceedings of SPE Technical Conference and Exhibition, 9–12 October, Dallas* (Society of Petroleum Engineers, Richardson, TX, 2005).
- [33] A. P. Sheppard, R. M. Sok, H. Averdunk, V. B. Robins, and A. Ghous, Analysis of rock microstructure using high-resolution x-ray tomography, in *Proceedings of the International Symposium of the Society of Core Analysts* (Society of Core Analysts, 2006).
- [34] A. Rabbani, S. Jamshidi, and S. Salehi, An automated simple algorithm for realistic pore network extraction from micro-tomography images, *J. Pet. Sci. Eng.* **123**, 164 (2014).
- [35] T. Agaesse, A. Lamibrac, F. N. Büchi, J. Pauchet, and M. Prat, Validation of pore network simulations of ex-situ water distributions in a gas diffusion layer of proton exchange membrane fuel cells with x-ray tomographic images, *J. Power Sources* **331**, 462 (2016).
- [36] F. Yang, F. F. Hingerl, X. Xiao, Y. Liu, Z. Wu, S. M. Benson, and M. F. Toney, Extraction of pore-morphology and capillary pressure curves of porous media from synchrotron-based tomography data, *Sci. Rep.* **5**, 10635 (2015).
- [37] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevE.96.023307> for the source code of the algorithm.
- [38] P. Soille, *Morphological Image Analysis—Principles and Applications* (Springer, Berlin, 2003).
- [39] M. J. Blunt, *Multiphase Flow in Permeable Media: A Pore-Scale Perspective* (Cambridge University Press, Cambridge, 2017).
- [40] E. Jones, T. Oliphant, and P. Peterson, SciPy: Open source scientific tools for Python (2001); available at <http://www.scipy.org/>.
- [41] G. Borgefors, Distance transformations in arbitrary dimensions, *Comput. Vision Graphics Image Process.* **27**, 321 (1984).

- [42] C. R. Maurer, R. Qi, and V. Raghavan, A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions, *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 265 (2003).
- [43] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, NIH Image to ImageJ: 25 years of image analysis, *Nat. Methods* **9**, 671 (2012).
- [44] J. Serra and L. Vincent, An overview of morphological filtering, *Circuits Syst. Signal Process.* **11**, 47 (1992).
- [45] S. Beucher and F. Meyer, The morphological approach to segmentation: The watershed transformation, in *Mathematical Morphology in Image Processing*, Optical Engineering Vol. 34, edited by E. Dougherty (Marcel Dekker, New York, 1992), p. 433.
- [46] J. Gostick, M. Aghighi, J. Hinebaugh, T. Tranter, M. A. Hoeh, H. Day, B. Spellacy, M. H. Sharqawy, A. Bazylak, A. Burns, W. Lehnert, and A. Putz, OpenPNM: A pore network modeling package, *Comput. Sci. Eng.* **18**, 60 (2016).
- [47] J.-W. Kim, D. Kim, and W. B. Lindquist, A re-examination of throats, *Water Resour. Res.* **49**, 7615 (2013).
- [48] X. Miao, K. M. Gerke, and T. O. Sizonenko, A new way to parameterize hydraulic conductances of pore elements: A step towards creating pore-networks without pore shape simplifications, *Adv. Water Resour.* **105**, 162 (2017).
- [49] J. T. Gostick, Random pore network modeling of fibrous PEMFC gas diffusion media using Voronoi and Delaunay tessellations, *J. Electrochem. Soc.* **160**, F731 (2013).
- [50] T. G. Tranter, A. D. Burns, D. B. Ingham, and M. Pourkashanian, The effects of compression on single and multiphase flow in a model polymer electrolyte membrane fuel cell gas diffusion layer, *Int. J. Hydrogen Energy* **40**, 652 (2015).
- [51] M. Hilpert and C. T. Miller, Pore-morphology-based simulation of drainage in totally wetting porous media, *Adv. Water Resour.* **24**, 243 (2001).
- [52] G. W. Jackson and D. F. James, The permeability of fibrous porous media, *Can. J. Chem. Eng.* **64**, 364 (1986).
- [53] M. M. Tomadakis and T. J. Robertson, Viscous permeability of random fiber structures: Comparison of electrical and diffusional estimates with experimental and analytical results, *J. Compos. Mater.* **39**, 163 (2005).
- [54] Imperial College Consortium on Pore-Scale Modelling. Berea Sandstone, 2014, doi:10.6084/m9.figshare.1153794.v2.
- [55] P. A. García-Salaberri, G. Hwang, M. Vera, A. Z. Weber, and J. T. Gostick, Effective diffusivity in partially-saturated carbon-fiber gas diffusion layers: Effect of through-plane saturation distribution, *Int. J. Heat Mass Transfer* **86**, 319 (2015).
- [56] P. A. García-Salaberri, J. T. Gostick, G. Hwang, A. Z. Weber, and M. Vera, Effective diffusivity in partially-saturated carbon-fiber gas diffusion layers: Effect of local saturation and application to macroscopic continuum models, *J. Power Sources* **296**, 440 (2015).
- [57] R. R. Rashapov, J. Unno, and J. T. Gostick, Characterization of PEMFC gas diffusion layer porosity, *J. Electrochem. Soc.* **162**, F603 (2015).
- [58] J. Gostick, M. Fowler, M. Pritzker, M. Ioannidis, and L. Behra, In-plane and through-plane gas permeability of carbon fiber electrode backing layers, *J. Power Sources* **162**, 228 (2006).
- [59] J. Gostick, M. Ioannidis, M. Fowler, and M. Pritzker, Pore network modeling of fibrous gas diffusion layers for polymer electrolyte membrane fuel cells, *J. Power Sources* **173**, 277 (2007).
- [60] I. V. Zenyuk, D. Y. Parkinson, L. G. Connolly, and A. Z. Weber, Gas-diffusion-layer structural properties under compression via x-ray tomography, *J. Power Sources* **328**, 364 (2016).
- [61] A. Lamibrac, J. Roth, M. Toulec, F. Marone, M. Stampanoni, and F. N. Buchi, Characterization of liquid water saturation in gas diffusion layers by x-ray tomographic microscopy, *J. Electrochem. Soc.* **163**, F202 (2016).
- [62] V. P. Schulz, J. Becker, A. Wiegmann, P. P. Mukherjee, and C.-Y. Wang, Modeling of two-phase behavior in the gas diffusion medium of PEFCs via full morphology approach, *J. Electrochem. Soc.* **154**, B419 (2007).
- [63] Y. Sholokhova, D. Kim, and W. Brent Lindquist, Network flow modeling via lattice-Boltzmann based channel conductance, *Adv. Water Resour.* **32**, 205 (2009).
- [64] M. L. Porter and D. Wildenschild, Image analysis algorithms for estimating porous media multiphase flow variables from computed microtomography data: A validation study, *Comput. Geosci.* **14**, 15 (2010).