InterPore JOURNAL

Original Research

MAGNET: MEDIAL AXIS GUIDED NETWORK EXTRACTION TOOL

Michael McKague, Hamed Fathiannasab, Mohammad Amin Sadeghi, Jeff Gostick

Department of Chemical Engineering, University of Waterloo, ON, Canada

Correspondence to:
Jeff Gostick,
jgostick@uwaterloo.ca

How to Cite:
McKague, M., Fathiannasab, H., Sadeghi, M., & Gostick, J. (2025). MAGNET: Medial Axis Guided Network Extraction Tool. *InterPore Journal*, 2(4), IPJ011225-7. https://doi.org/10.69631/g47x8w91

RECEIVED: 27 May 2025
ACCEPTED: 2 Aug. 2025
PUBLISHED: 1 Dec. 2025

ABSTRACT

Pore network models are useful for studying transport in porous materials in a computationally efficient way. Extraction of networks from volumetric images has evolved over the years, starting with medial axis-based approaches to more recent watershed segmentation. This paper reconsiders the classic medial axis method, which offers several advantages such as speed and topological correctness, and develops a modernized, updated, and improved version. The new method is named Medial Axis Guided Network Extraction Tool (MAGNET). It works by analyzing the skeleton of a porous material to identify pore centers at junctions and endpoints. Additional pore bodies are found on long throats using two different approaches. This work includes an efficient tool for calculating the cross-sectional area of throats with irregular shape by using walkers with an infinite mean-free path to probe the geometry orthogonal to the medial axis at the point of the throat constriction. This extra step was critical for obtaining an equivalent diameter needed to calculate the permeability. Lastly, MAGNET was written with computational efficiency in mind. The skeletonization approach was itself 4.2X faster than the SNOW watershed segmentation for a 1000^3 image. Additionally, a parallelized skeletonization was applied by processing the image in blocks with sufficient overlap which resulted in a 5.5X speed-up compared to the serial approach. To validate the output, MAGNET was tested on a 400^3 voxel image of a Berea sandstone, and the flow and capillary properties of the extracted network were compared to the results from SNOW and the lattice-Boltzmann method. Structural information such as pore and throat size distribution and mercury intrusion curves was compared, and noticeable similarity was achieved. Crucially, the permeability predicted by MAGNET was within 5% of the lattice-Boltzmann prediction on the same image.

KEYWORDS

Network extraction, Pore network modelling, Image processing, Permeability, Porous media

1. INTRODUCTION

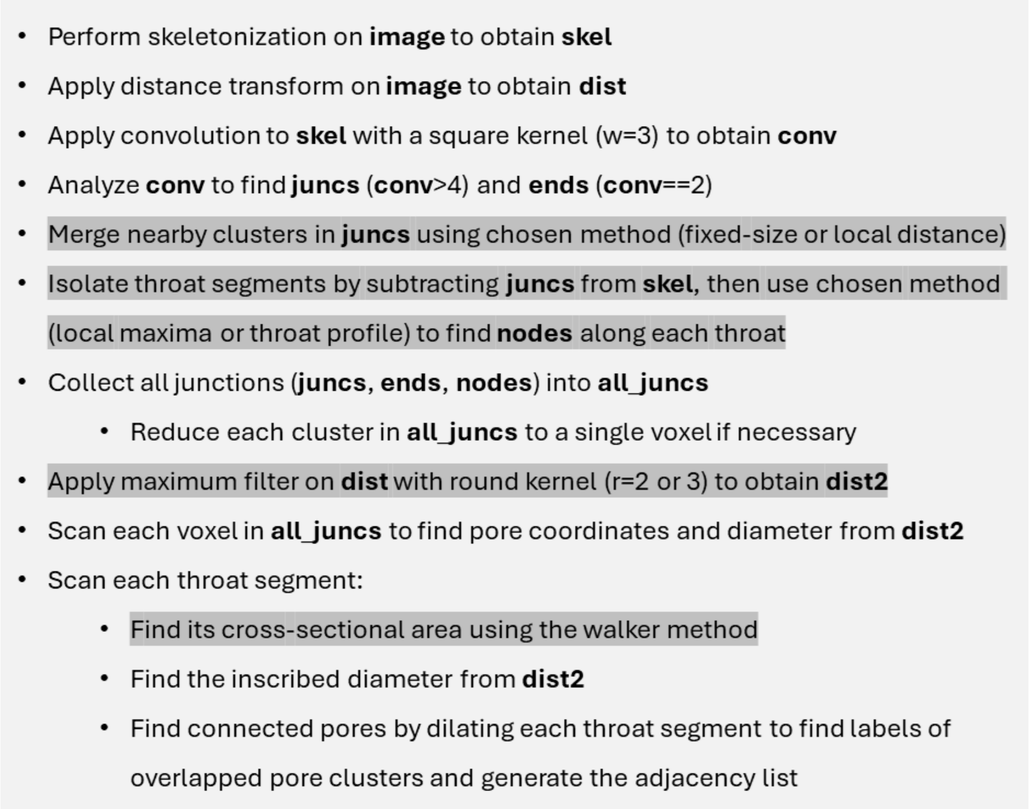
Advances in imaging tools such as x-ray computed tomography and focused ion beam milling coupled with scanning electron microscopy (SEM) imaging has enabled the study of porous materials at the micron scale or smaller (22) via analysis of and simulations on voxel images of the microstructure (28). Because the starting point is a volumetric image, the acquisition of topological, geometrical, and transport properties is based on image processing algorithms. Using volumetric images, it is possible to estimate tortuosity (17), coordination number distributions (6, 15, 33), and pore and throat size distributions (2, 15, 19, 33), among others. One particularly useful but challenging task is the extraction of pore networks (37). Pore networks combine both topological and geometrical information to create a simplified representation of the void space which can then be used to simulate transport processes such as permeability and tortuosity, as well as multiphase conditions, with excellent computational efficiency. Increasing the speed and accuracy of pore network extraction algorithms is an ongoing pursuit (12).

Over the years, various network extraction algorithms have been developed, which can generally be grouped into three different categories: maximal ball (6, 31), watershed segmentation (8, 26, 29, 34), and medial axis (11, 15, 17). The maximal ball approach works by inserting the largest inscribed sphere at each void voxel. Spheres that are fully encompassed by larger ones are removed, leaving the so-called "maximal balls" (6). The smaller of the "maximal balls" situated between relatively larger "maximal balls" are considered throats while those adjoining larger "maximal balls" are considered pores. One drawback of this approach is the time-consuming process of interpretating which balls belong to which pores (8, 27). Watershed segmentation works by segmenting the void space into regions corresponding to pores. This method was first explored by Thompson et al. (34) and Sheppard et al. (29). More recently, detailed validation has been provided by Rabbani et al. (26) and Gostick et al. (8), and an open-source implementation is now available (7). This algorithm works by identifying local peaks in the distance transform, removing saddle points and nearby peaks to avoid over-segmentation, and finally applying the watershed algorithm to segment the image into pore regions (8). One of the main disadvantages of this approach is the computational effort of the watershed filter, though the process of correctly identifying peaks and therefore pore bodies is not entirely reliable as it involves several adjustable parameters. The third and final approach is based on the medial axis. The medial axis, also often referred to as the skeleton of an object, is defined by the set of points that are equidistant to two or more points on the object border (3). The skeleton retains all the topological information of the original image, which is one of its main appeals. When combined with a distance transform, the skeleton can also be used to obtain geometrical data such as pore and throat diameters by looking at distance values corresponding to voxels on the skeleton. Lindquist et al. (17) were the first to explore the use of the medial axis as a tool for analyzing the geometry of a porous material, and this was developed into a software implementation called 3DMA-Rock (16). The advantage of this approach over the other two methods is its simplicity owing to the direct relationship between the topological skeleton and the pore network, as well as the potential efficiency since skeletonization is comparatively fast.

The aim of the present paper is to revisit and revise medial axis-based network extraction with the following aims: **a)** to create a computationally efficient and accurate network extraction tool, and **b)** to provide a modernized, user-friendly, and open-source implementation. Past developments of a medial axis-based network extraction were developed in either MATLAB (15) or FORTRAN (16), however the code was not widely distributed and required significant technical know-how of the user. Our new algorithm, which is referred to as Medial Axis Guided Network Extraction Tool (MAGNET), was written in Python, a popular and easy to use programming language, and is included in the open-source package PoreSpy which is publicly available on Github and distributed via the Python Package Index (PyPI). Aside from being more accessible and modernized, MAGNET was also written to be more efficient than commonly used options such as the watershed segmentation approach developed by Gostick et al. (8). As will be shown, MAGNET is also faster than the recently presented PREGO algorithm (13) which uses region-growing in place of the standard watershed. To fully capitalize on MAGNET's inherent speed, the present work also includes a study on the use of image chunking to apply skeletonization in parallel, since skeletonization is the slowest step.

In order to provide an accurate network, this work addresses some of the challenges faced by medial axis network extraction algorithms. Firstly, dealing with overlapping pores can be complicated, since merging criteria must be defined, the details of which can vary between implementations (25). Secondly, determining pore and throat sizes from the skeleton can be incorrect since the position of the skeleton does not always coincide with the actual maxima of the distance transforms. Moreover, using the distance transform to find pore/throat sizes means that only the smallest dimension is known. This is problematic for pores/throats with an aspect ratio above unity, since both sizes control the transport and capillary properties. Lastly, pores with a connectivity of 2 are misclassified as long throats, which can lead to errors when computing flow since the smallest diameter between pores will be applied to the entire length. The algorithm developed herein addresses all these problems in novel ways that are simple enough to be efficient, while still effectively remedying the problem.

The present work uses a pore-finding approach, as opposed to a throat finding approach (25), which works by locating junctions and terminal points (or endpoints) in the skeleton. Nearby junctions are merged if they are close but otherwise they are allowed to overlap which can be accommodated in the pore network modeling software using appropriate conductance models. This work explores two different methods to identify pores with a connectivity of 2, by either looking for local maxima along throat segments or by analyzing the throat profile for peaks. Also, it was found that applying a maximum filter was sufficient to correct the fact that the skeleton does not always lie on the true maxima in the distance transform. Finally, and most importantly, this work proposes a new approach to finding area and shape of throat openings. Previous efforts have addressed this by slicing (15) or dilating (18) subsections of the volumetric image, but in this work a procedure is proposed that uses walkers with an infinite mean-free path (36), travelling orthogonally away from the skeleton in all directions, to probe the shape of the local geometry. As will be shown, MAGNET is both faster than comparable network extraction tools and quite accurate at predicting transport properties and capillary pressure behavior, providing a modern and improved implementation of a classic algorithm.



```
• Perform skeletonization on image to obtain skel
• Apply distance transform on image to obtain dist
• Apply convolution to skel with a square kernel (w=3) to obtain conv
• Analyze conv to find juncs (conv>4) and ends (conv==2)
• Merge nearby clusters in juncs using chosen method (fixed-size or local distance)
• Isolate throat segments by subtracting juncs from skel, then use chosen method
  (local maxima or throat profile) to find nodes along each throat
• Collect all junctions (juncs, ends, nodes) into all_juncs
    • Reduce each cluster in all_juncs to a single voxel if necessary
• Apply maximum filter on dist with round kernel (r=2 or 3) to obtain dist2
• Scan each voxel in all_juncs to find pore coordinates and diameter from dist2
• Scan each throat segment:
    • Find its cross-sectional area using the walker method
    • Find the inscribed diameter from dist2
    • Find connected pores by dilating each throat segment to find labels of
      overlapped pore clusters and generate the adjacency list
```

Figure 1: Pseudo-code for MAGNET. The steps highlighted in dark gray indicate new approaches introduced in this work.

https://doi.org/10.69631/g47x8w91

2. ALGORITHM

Figure 1 is a flow-chart of the image processing steps used by MAGNET. The steps highlighted by dark gray indicate where new approaches have been developed as part of this work. It starts by obtaining the skeleton and the distance transform of the void space which are used repeatedly over the course of the network extraction. Next, convolution is applied to the skeleton to find all junctions and endpoints followed by merging. After merging, the clusters of junctions are removed from the skeleton, and the remaining skeleton segments are searched for additional pores with a connectivity of 2. Pore and throat sizes are found by looking at the distance transform values underlaying each identified junction. Crucially, when obtaining throat sizes, a new algorithm was developed which can extract the area of throats and not just their inscribed diameter. Finally, the connectivity of the network is determined from the image, then all properties are combined into a Python dictionary for easy conversion to OpenPNM[a] for performing network simulations, though the data format is general enough to be used with any simulation tool.

2.1. Skeleton

The process of obtaining the skeleton is sometimes called "image thinning" (9) or "ultimate erosion", and the term skeleton and medial axis are used interchangeably. There are generally two main approaches to image thinning algorithms: kernel-based filters and decision trees. Zhang et al. developed the first algorithm for image thinning using a kernel-based approach that iterates over each pixel in an image with sub-iterations to remove contour pixels that do not lie on the skeleton (39). This approach, however, was only designed for 2D images. Lee et al. developed an image thinning algorithm that effectively reduces 3D structures to their skeletal forms by iteratively removing surface layers of voxels while preserving the essential topology (14). The algorithm is based on a decision tree approach that evaluates each voxel's connectivity within its local neighborhood. At each iteration, it identifies and removes "simple" surface voxels—those that, when removed, do not alter the topology or connectivity of the structure. Given the importance of volumetric images in the study of porous media, Lee's method

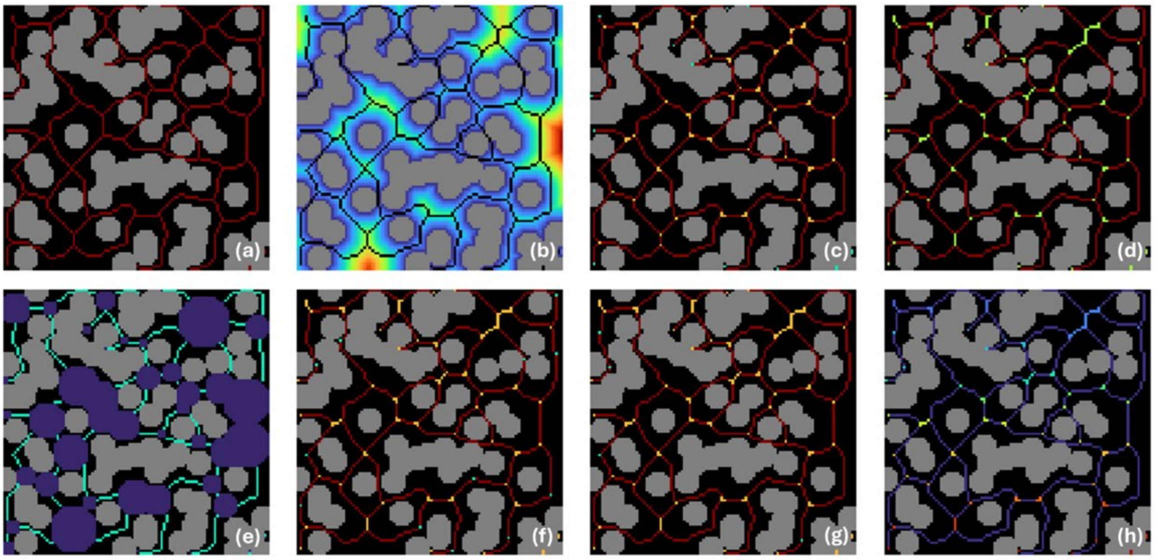


Figure 2: Progression of MAGNET steps. (a) Lee's skeleton on a 100 by 100 image of packed spheres with radius of 5 pixels and 60% porosity. (b) is the distance transform with skeleton overlaid. (c) shows the junctions (orange) and endpoints (cyan) found from convolution on the skeleton. (d) shows clusters of junctions and endpoints (green) that form after merging. (e) shows spherical pore bodies inserted at junctions and endpoints (purple) used for identifying the segments of the skeleton that do not overlap a pore body (cyan). (f) shows the throat nodes (cyan) found via the maximum filter method. (g) shows the throat nodes (cyan) found using the throat profile method. (h) shows the entire set of nodes color-coded by their pore index number, along with the throat segments (purple).

[a] https://openpnm.org/

has been used extensively by porous media researchers including in the original medial axis network extraction by Lindquist et al. (17). Lee's method is publicly available as part of scikit-image[b]. **Figure 2a** shows Lee's skeleton applied to a 100 by 100 image of packed spheres with radius of 5 pixels and 60% porosity. By default, MAGNET uses the scikit-image implementation for determining the skeleton, though it can work with any skeleton so in principle an alternative skeletonization algorithm could be used. However, it should be stressed that the skeleton must be a single pixel thick for pore identification to work properly. One alternative thinning algorithm worth mentioning for applications in porous media is that of Delgado-Friedrichs et al., who present a state-of-the-art skeletonization approach based on Discrete Morse Theory (5).

Because of their relevance to porous materials, there have been some image thinning algorithms developed specifically for this use. Thovert et al. developed an image thinning algorithm and applied it to stochastic images of porous media, but could not guarantee connectivity of the pore space (35). Connectivity is important in preserving the topology of the original structure (23). For this reason, Liang et al. (15) elected to use Ma and Sonka's thinning algorithm (20) which guaranteed connectivity. Palagyi and Kuba (24) discuss the importance of preserving the topology, as they present a thinning algorithm for 3D objects that do not form medial surfaces. The skeleton provides not only topological information but can also be multiplied by the distance transform to obtain the medial axis transform, which in turn can be used to determine geometrical properties as well the size of spheres that can be inserted at each point.

Prior to obtaining the skeleton there are several necessary steps that should be taken to clean-up the original image. Firstly, "floating solids" must be removed since the skeleton will form a "shell" around them. In 2D this is not a problem since the skeleton remains an 'axis'; however, in 3D this leads to "surfaces". This is illustrated in **Figure 3** where a shell, shown in purple, forms around a floating solid, shown in blue. While floating solids are physically impossible, they are common in tomography images following improper binarization of the original greyscale image. This can be easily avoided by pre-processing images to remove solid that is not connected to the bulk solid phase prior to determining the skeleton (10). This is done automatically by MAGNET but should be noted for users working with their own skeleton. In the case that the user provides MAGNET with their own skeleton (e.g., from ImageJ or other image processing application), a check is performed to ensure that it contains no hollow shells. Secondly, it is also helpful to fill blind pores since these lead to numerical errors when pore network simulations are run. Although these can be removed from the network later, it is convenient to prevent their existence in the first place.

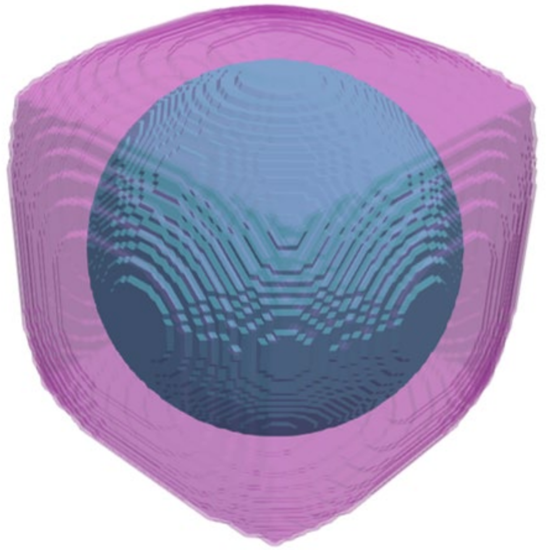


Figure 3: Skeleton (purple) forming a shell on a floating sphere (blue).

Finding the skeleton of an image can represent substantial computational costs on the total network extraction time. In fact, MAGNET can spend almost 30% of its computational time on skeletonization (see **Fig. 12**). Therefore, to improve efficiency, the "block-and-tackle" approach was explored, whereby the image was subdivided into blocks, with some overlap, and skeletonization was performed on these blocks in parallel (using dask). To quantify the speed-up of this approach, four artificial images of overlapping spheres with a radius of 10 voxels and 50% porosity were generated using PoreSpy with a random seed value of 10. Each image was assigned a different shape, but each shape had the same pattern of (n, n, 2n) with values of n being 250, 500, 1000, and 1500 for the four

[b] *https://scikit-image.org/*

images. This shape pattern allowed for the image to be divided into 16 blocks using divisions (2, 2, 4) with each block having the same size. Then setting the number of cores to 16, the skeleton was parallelized across 16 logical cores which were available on the workstation used for this analysis.

The main challenge with subdividing the image was determining the amount of overlap to use between image blocks to ensure a correct result. The amount of overlap was studied by trying different values on the four images and comparing the resulting skeleton to the true skeleton obtained using the serial method. **Figure 4a** shows the effect of overlap between blocks on the four images of overlapping spheres. The accuracy of the parallel skeleton was calculated by counting the number of voxels shared by both serial and parallel skeletons and then dividing that number by the total size of the serial skeleton. The results showed that 100% accuracy was reached at 60, 72, 72, and 78 voxels respectively for each of the four images increasing in size. The maximum distance transform of the four images, also in order of increasing size, were 20, 25, 27, and 31. Therefore, a general estimate for overlap of image chunks could be ~2-3 times the maximum distance transform value of the image.

After determining the correct amount of overlap for each image size, the computational time of the parallel skeleton was compared to the serial skeleton. **Figure 4b** shows the resulting comparison for each of the four images tested, and it was observed that the speed up increased with the size of the image. The (250, 250, 500) sized image experienced a speed up of only 2 times while a maximum amount of speed up of 5.5 was observed for the (1500, 1500, 3000) sized image. This observation is an effect of the diminishing ratio of overlapped voxels to total voxels as the image size increased. For example, the number of extra voxels due to overlapping for the smallest image was calculated to be 5.25 x 10^7, however the image itself only had a total of 3.125 x 10^7 voxels. Here, the extra number of voxels due to overlapping exceeded the original number of voxels. This means that more than twice the original number of voxels was being distributed across 16 logical cores, explaining why only moderate speedup was achieved. However, the extra number of voxels required for overlapping in the largest image was just 36% of the original size. Therefore, less time was spent on these overlapping regions, and more time was spent on the actual image resulting in a higher speedup. For this reason and based on the results from this investigation, parallelization of the skeleton is recommended only for large images, or images with small pores, where a proportionally small amount of overlap is required. This is also justified by the fact that computational time is not an important consideration if the image is smaller.

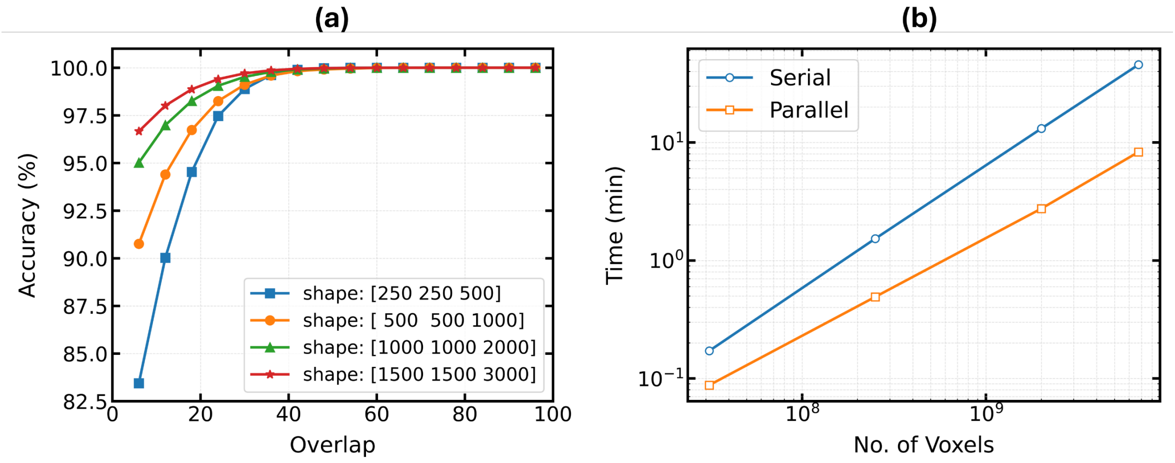


Figure 4: (a) The effect of overlap between image chunks on the resulting accuracy of the parallelized skeleton compared to the serial skeleton. A range of overlaps from 6 to 96 voxels incrementing by 6 voxels was tried on 4 different images of shape (*n*, *n*, 2*n*). The image was divided by 2, 2, 4 in the x, y, and z directions respectively resulting in an even 16 image chunks for parallel operation on 16 cores. **(b)** The resulting computation times of the parallel skeleton (orange) compared to the serial skeleton (blue). The skeleton, serial and parallel, was taken on four images of overlapping spheres with radius of 10 voxels and porosity of 50%. The shapes of the four images were: [250, 250, 500], [500, 500, 1000], [1000, 1000, 2000], and [1500, 1500, 3000]. The results showed increasing speedup with increasing size ranging from 2 for the smallest image and 5.5 for the largest image.

https://doi.org/10.69631/g47x8w91

A remaining challenge with the skeleton is locating boundary or surface pores (25) as the normal skeleton does not extend to the edge of the image (see **Fig. 2a**). Attempts were made to coerce the skeleton to the edge of the image using strategies based on padding the image prior to skeletonization and trimming the padding thereafter. However, while this idea worked successfully in 2D, ultimately all methods resulted in a skeleton that was fragmented and clustered for 3D images. For this reason, boundary pores are not labelled by MAGNET and the user is left to find or add boundary pores using their own criteria for simulations requiring them. The simulations in this work assumed boundary pores to be within 2% of the edge of the image boundary. Developing a skeletonization algorithm that correctly extends beyond the edges of the images remains a valuable target for future research.

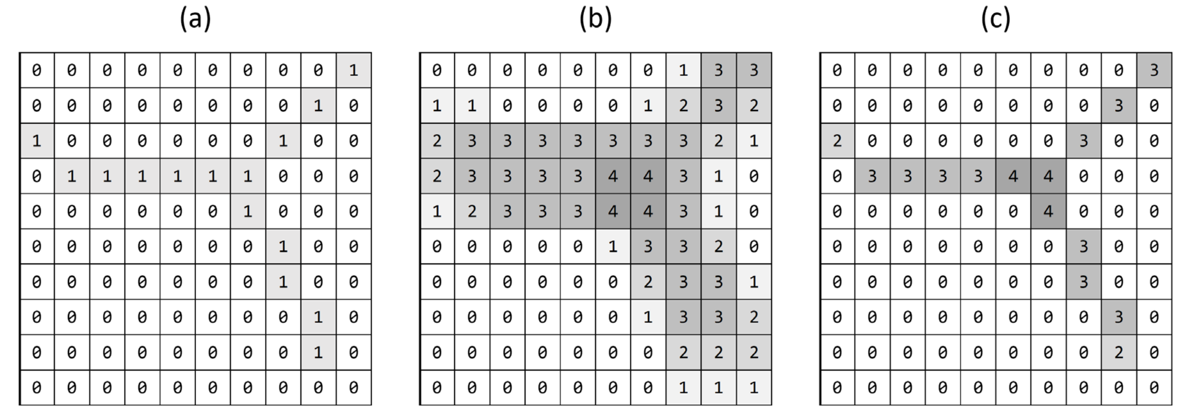
2.2. Distance Transform

The distance transform is a standard tool used in image processing of porous media. It replaces each foreground voxel with the Euclidean distance to the nearest background voxel and, therefore, is useful for obtaining the geometrical properties of pores and throats. Computing the distance transform can be computationally intensive; however, the open source `edt` package (32) was used here, which is efficient and is also parallelized.

Figure 2b shows the resulting distance transform on the void space, with the skeleton overlaid in black. The skeleton tends to follow the contour of the brightest pixels, which have the largest Euclidean distance; however, this is not always guaranteed. One example is at the edge of the image where the skeleton remains approximately centered between the edge of the image and the neighboring solid phase while the distance transform continues to measure from the nearest solid. The failure of the skeleton to lie on the true maximum of the distance transform also occurs within the center of the image, but to a less noticeable extent. For example, on a 256^3 image of blobs with 70% porosity, it was found that only 13.6% of peaks in the distance transform map were located on the skeleton. This issue means that the pore and throat sizes, which are taken from the distance transform values underlying the skeleton are generally biased toward small sizes. This is why it is proposed in **Figure 1** to take the maximum filter of the distance transform using a small kernel prior to obtaining pore and throat sizes as discussed further in Section 2.6.

2.3. Find Junctions and Endpoints

The next step is to analyze the skeleton for junctions and endpoints, which will become pore centers, by applying a convolution to the skeleton. The convolution replaces each voxel with the number of skeleton voxels in the vicinity of a cubic kernel 3 voxels across. **Figure 5** shows the process in 2D by looking closely at a 10 by 10 area of pixels representing a subimage of a larger two-dimensional skeleton. Note that the left hand side of the subimage represents the edge of the larger image whereas all other sides in the subimage do not. **Figure 5a** is the original skeleton prior to convolution and **Figure 5b** is the resulting image after convolution. Following convolution, the image is masked using the original skeleton image to yield **Figure 5c**. All pixels with a value of 4 or greater are junction points, whereas



(a)

0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1	0	0
0	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0

(b)

0	0	0	0	0	0	0	1	3	3
1	1	0	0	0	0	1	2	3	2
2	3	3	3	3	3	3	3	2	1
2	3	3	3	3	4	4	3	1	0
1	2	3	3	3	4	4	3	1	0
0	0	0	0	0	1	3	3	2	0
0	0	0	0	0	0	2	3	3	1
0	0	0	0	0	0	1	3	3	2
0	0	0	0	0	0	0	2	2	2
0	0	0	0	0	0	0	1	1	1

(c)

0	0	0	0	0	0	0	0	0	3
0	0	0	0	0	0	0	0	3	0
2	0	0	0	0	0	0	3	0	0
0	3	3	3	3	4	4	0	0	0
0	0	0	0	0	0	4	0	0	0
0	0	0	0	0	0	0	3	0	0
0	0	0	0	0	0	0	3	0	0
0	0	0	0	0	0	0	0	3	0
0	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	0

Figure 5: A close up view of a 10 by 10 pixel area of a skeleton **(a)**, skeleton following convolution **(b)**, and the final convolution after the skeleton mask **(c)**.

pixels with a value of 2 are endpoints. Endpoints could exist at the image edge as shown on the left hand side of **Figure 5c** or in the internal image. However, endpoints will rarely exist, if at all, at the image edge unless otherwise coerced by some means.

Figure 2c shows the result of finding junctions and endpoints using convolution on the skeleton taken from the image of overlapping spheres introduced above. Note that the convolution operation can result in clusters of junctions where several branches nearly coincide. Liang et al. devised a sophisticated strategy to eliminate clustering (15), but MAGNET handles clustering more gracefully by simply merging junctions into a single cluster, and therefore a single pore, as shown by the green pixels in **Figure 2d**. Junctions can be merged by two different options: by applying either a fixed threshold or by using an adaptive threshold based on the local distance transform values. In both cases, the process starts by removing the junctions and endpoints from the image, leaving only the connecting segments. For the fixed threshold, segments are re-added to the image if their number of voxels is smaller than a specified threshold (e.g. 3 pixels). This will cause junctions that are 3 voxels apart to merge into a single junction. **Figure 2d** shows the result of using a fixed threshold of 3 pixels, with the section in the top-right illustrating clearly how several junctions are merged into a single one. For the adaptive distance-based threshold, segments are re-added to the image if their number of voxels is smaller than the maximum of the distance transform underlying the segment. This allows junctions in larger pores to become connected even if they are further apart than junctions in small pores. In either case, after the junctions are merged, it is necessary to choose a single voxel from each cluster which will be deemed the center point of the pore. The voxel with the largest distance transform value is chosen. In the event that multiple voxels have the same value, then a random one is chosen from them.

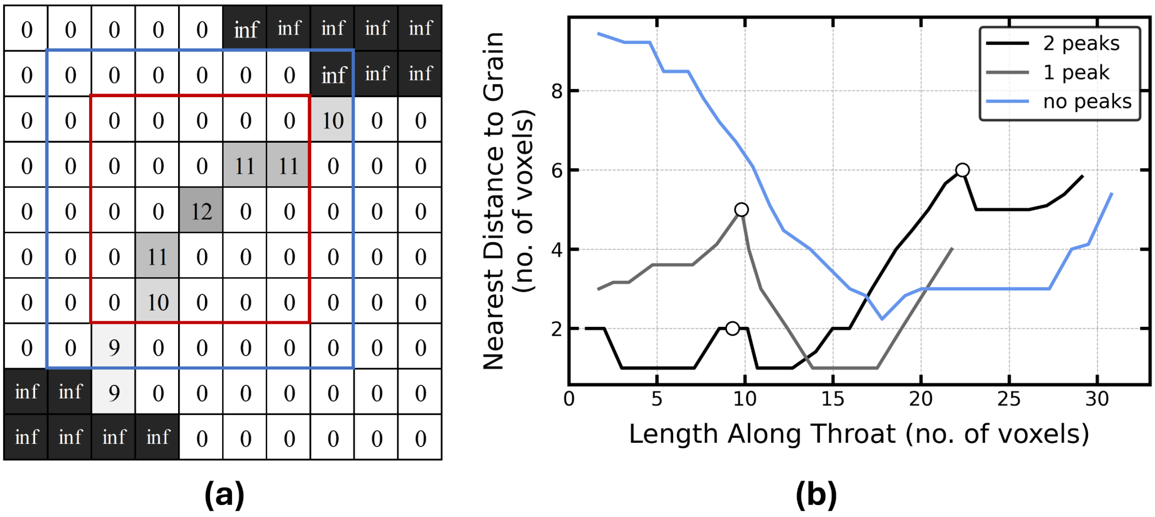
2.4. Finding Throat Junctions

A limitation of finding pores by looking for junctions is that it misidentifies pores with a coordination number of 2 as being part of a long throat. This can be problematic in some applications as it will result in long throats which can cause difficulties when modelling reactions since pore network models generally assume that reactions occur in pores. Therefore, this work presents two alternative approaches for finding “throat junctions”. The two approaches presented are **1)** using a maximum filter to find local maximum on each skeleton segment and **2)** using the fast-marching method to sort the voxels of each segment into a linear chain, then using signal processing tools to find peaks in the profile using the underlying distance transform value as the heights. It is important to stress that these two methods are not intended to be topologically consistent but two fundamentally different ways to characterize junctions along throats. The two approaches are explained in detail below.

2.4.1. Maximum Filter

To find throat junctions using a maximum filter, pore bodies are first inserted as spheres at the pore centers found from junctions and endpoints in the previous step. **Figure 2e** shows an example of disks inserted at pore centers in the image of overlapping spheres. The radius of the inserted spheres (or disks in **Fig. 2e**) is equal to the value of the distance transform at the pore center explaining why each disk in **Figure 2e** just touches the nearest solid. Unfortunately, inserting pore bodies into an image using Python for loops can be slow for a large number of pores. Therefore, MAGNET uses just-in-time compilation, using the Numba package, to speed up for loops and reduce computation time.

After inserting spheres, the remaining skeleton segments (cyan color in **Fig. 2e**) are searched for local maximums in the distance transform. To ensure that the ends of the segments are not inadvertently identified, the distance transform value in the voxels belonging to the pore bodies are set to infinity. **Figure 6** is a 10-pixel by 10-pixel area of the resulting image, showing a skeleton connecting two pore bodies. The connected pores have values of infinity as shown in the top right and bottom left corners while the skeleton is labelled according to its distance to the nearest solid (aka the medial axis transform). Finally, the maximum filter is applied to this image to find maxima along the remaining skeleton. Any pixels which retain their value after applying the maximum filter are a local maximum. The resulting local maximums are the throat junctions shown in **Figure 2f** as cyan pixels.



0	0	0	0	0	inf	inf	inf	inf	inf
0	0	0	0	0	0	0	inf	inf	inf
0	0	0	0	0	0	0	10	0	0
0	0	0	0	0	11	11	0	0	0
0	0	0	0	12	0	0	0	0	0
0	0	0	11	0	0	0	0	0	0
0	0	0	10	0	0	0	0	0	0
0	0	9	0	0	0	0	0	0	0
inf	inf	9	0	0	0	0	0	0	0
inf	inf	inf	inf	0	0	0	0	0	0

Figure 6: (a) A 10 by 10 array of pixels showing a skeleton connecting two inserted pore bodies. The inserted pores (bottom left and top right) are labelled with the value infinity while the skeleton is labelled with size values corresponding to the value of the distance transform. Local maximums are searched for along the remaining skeleton using the maximum filter, but different sized neighboring regions can be used. The red and blue bounding boxes represent a neighboring region of 5 pixels and 7 pixels wide respectively, centered on the local maximum. **(b)** The peak profile along the length of throats with 2 peaks (black), 1 peak (grey), and no peaks (blue) where peaks are identified as circle markers. The distance to the nearest grain and the distance along each throat is in number of voxels.

The size of the neighboring region in applying the maximum filter is an adjustable parameter. By default, MAGNET uses a size of 7 voxels meaning that the neighboring region is defined as three voxels in each direction from the center of the kernel element. To illustrate this, red and blue bounding boxes representing 5- and 7-pixel long neighboring regions centered at the local maximum (i.e. 12) were added to **Figure 6a**. In the case of the smaller footprint, the local maximum along the skeleton is found whereas the larger footprint overlaps with the adjacent pore body, and therefore, does not find the local maximum on the skeleton since pore bodies were labelled with infinity. This illustration shows how the kernel size effectively controls which throats are classified as long and should be broken into 2 throats and a pore (at least).

2.4.2. Throat Profile

The other approach for finding throat junctions looks for peaks in the profile of the distance values along each segment. To create a profile, however, it was necessary to determine the order of the voxels along the segment from one pore to the neighbor. This was accomplished using the fast-marching method, which starts at one end and steps along the segment, labeling each voxel with the distance travelled, until reaching the last voxel on the other end. A profile can then be created by plotting the distance travelled vs. the distance transform value for each voxel. **Figure 6b** shows the peak profile for three throats, with 2 peaks, 1 peak, and no peaks. The peaks were identified using a circle marker and are found using the *find_peaks* function in Scipy's signal processing toolbox. This function accepts numerous arguments to ensure only valid peaks are found, such as avoiding peaks which are too close together and avoiding peaks which do not stand high enough relative to the neighboring voxels. These are adjustable, but in the present work only peaks with a relative height of 1 voxel or more and a spacing greater than the minimum distance value along the profile were accepted. **Figure 2g** shows the resulting image of junctions on the image of overlapping spheres after adding throat junctions. The throat with two junctions in **Figure 6b** is visible in the top left corner of **Figure 2g** for instance.

2.5. Determining Throat Area

While the distance transform is an important tool in the study of porous media for storing geometrical data, it is limited in that it only reports the distance to the nearest grain voxel. In real porous media, throat cross-sections can take on a variety of irregular shapes with varying aspect ratios. For example, consider the simple case of an elliptical cross-section, as shown in **Figure 7a**. Here, the distance transform is not sufficiently useful for calculating the throat area because it only contains the distance of the minor axis (or shorter radius, r_1) and knows nothing about the length of the major axis (or larger radius, r_2). The distance transform will, therefore, always underestimate the throat area because it reports only the shortest distance. This problem has been faced by other medial axis network extractions in the past, and a variety of different solutions have been proposed (15, 18, 25, 30). In general, there have been two different ways of finding throat area and these can be classified, by our own recognition, as either throat slicing or throat dilating. The work by Liang et al., is an example of using throat slicing to find the cross-sectional area (15). In that work, three consecutive voxels on the skeleton were used to find the vector normal to the plane of cross-section. Then, the image was sliced according to the plane of cross-section and the throat area estimated by counting the number of voxels. Alternatively, the skeleton can be dilated to find the cross-sectional area of the throat. Lindquist et al. did this by dilating the skeleton in a radial direction and stopping at points that touch the grain but continuing in other directions until eventually a closed loop forms representing the minimum throat surface area (18). Both approaches, however, become computationally expensive for a large number of throats.

In this work, random walk simulations which are well suited to voxel images and commonly used to estimate properties such as directional tortuosity (36), have inspired a novel approach for finding the throat area that uses walkers. While this approach might be similar to how throat cross-sections were found in the work by Bakke and Oren (4), it's unclear how the distance to the nearest solid was found exactly, and our technique proposes the use of the distance transform for faster computation. In our work, walkers are used to find the throat area by setting walkers to march out radially from the skeleton to the grain boundary, recording the distance each walker takes to reach the grain, and using triangulation to approximate the throat area. The direction taken by each walker is in a plane normal to the skeleton with equally spaced angles between adjacent walkers spanning a full revolution around the skeleton. To find the direction each walker takes, the dot product is taken with the plane's normal and set equal to zero. This results in **Equation 1** which constrains the resulting set of directional vectors, defined by angles φ_d and θ_d in spherical coordinates, to directions that lie in the plane perpendicular to the skeleton. The angle θ_d spans the full revolution around the skeleton from zero to 2π with even intervals of $2\pi/n$ where n is the number of walkers. The plane's normal is found using three consecutive

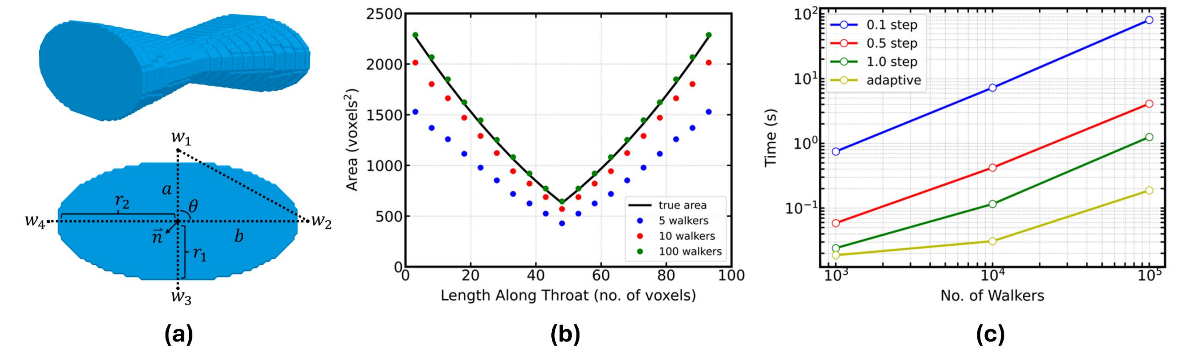


Figure 7: Digitally generated throat for demonstrating how walkers can be used to measure the cross-sectional area of a throat. Image **(a)** is a full view of the digitally generated throat while image **(b)** shows the cross-section and the path four walkers (w_1, w_2, w_3, and w_4) at directions 90 degrees from each other would take to estimate the area. **(b)** Measured area across length of digitally generated throat using 5 (blue), 10 (red), and 100 (green) walkers with a step size of 0.5 voxels. The black line represents the true area calculated using the equation for area of an ellipse. **(c)** The computational time to estimate throat area using different step sizes and number of walkers. The step sizes the walkers took were 0.1, 0.5, and 1.0 voxels as shown in blue, red, and green respectively. The yellow line shows an adaptive stepping scheme that uses the distance transform for which significant speed up was observed.

voxels in the skeleton, similar to the work by Liang et al. (6), and is expressed in spherical coordinates to obtain angles φ_{n} and θ_{n}.

$$\tan(\varphi_{\mathrm{d}})\tan(\varphi_{\mathrm{n}})\cos(\theta_{\mathrm{n}}-\theta_{\mathrm{d}})=-1 \tag{1}$$

After calculating the direction each walker is to take, the starting positions of the walkers are set and the *x*, *y*, and z positions are recorded. Then, the walkers march out from the skeleton at certain step sizes and their cartesian coordinates updated. Upon each step, walkers are checked to see if they land in a grain voxel at which point the walker stops walking. The walking procedure continues until all walkers reach the grain boundary. Prior to walking, the image is padded with grain voxels to keep walkers from marching outside of the image boundary. The final coordinates of the walkers can be compared to their starting positions to calculate the distance traveled. **Figure 7a** shows a set of four walkers (w_1, w_2, w_3, and w_4) marching out from the skeleton in a plane perpendicular to the normal vector and spanning a full revolution around the skeleton with an angle of $\frac{\pi}{2}$ between walkers. A triangulation, similar to the work by Shin et al. (30), was used to estimate the throat area. In **Figure 7a**, the dashed line connecting the end positions of walkers 1 and 2 encloses the estimated area as a triangle where lengths a and b are the distances travelled by walkers 1 and 2 respectively. **Equation 2** can be used to calculate the area of any triangle connecting the path taken by two walkers where θ is the angle between walker trajectories. The area between adjacent walkers, can be estimated in this way (using triangles), and the resulting area can be summed together to get a throat area.

$$A=\frac{ab}{2}\sin(\theta) \tag{2}$$

Figure 7b shows the effect of the number of walkers on estimating the cross-sectional area along the length of the digital throat in **Figure 7a** as an example. Walkers were sent out from the medial axis of the generated throat by trying 5, 10, and 100 walkers with a step size of 0.5 voxels. The distance the walkers travelled to reach the grain boundary was measured and triangulation was used to estimate the cross-sectional area. The calculated area along the length of the throat is plotted in **Figure 7b** along with the true area (in black) which was calculated using the equation for the area of an ellipse. The triangulation of the throat area is underestimated when using 5 walkers, a good approximation when using 10 walkers, and almost identical to the actual area for 100 walkers.

The step size taken by the walkers is an important factor in the computational time required. If the step size is too large (e.g. 5 voxels) then walkers end their path too far inside the solid, creating a bias towards longer paths. If the step size is too small (e.g. 0.1 voxels), then the result will be more accurate but more time consuming. **Figure 7c** shows the effect of step size on the computational time. This study was run on a workstation with an Intel(R) Xeon(R) CPU E7- 4860 processor with a speed of 2.27 GHz and 748 GB of RAM. Step sizes of 0.1, 0.5, and 1.0 voxel were tried for all walkers as well as an adaptive stepping scheme. The adaptive stepping works by having walkers step an amount equal to the distance transform underlying the voxel in which they are currently located. The idea is that walkers are never closer to the grain boundary than the value indicated in the distance transform, and therefore it is safe for the walker to travel as much as that distance in any direction without overstepping into the grain boundary. The results from measuring the computation time showed that a million walkers took roughly 78 minutes 41 seconds, 3 minutes 52 seconds, 1 minute 11 seconds, and 7.8 seconds for the 0.1, 0.5, 1.0 voxel, and adaptive step respectively. A million walkers in a pore network are reasonable considering that this is equivalent to 10 walkers per throat in a network with 100,000 throats. The speed up achieved from taking a 1 voxel step to using the distance transform for adaptive stepping was 9.1X faster.

2.6. Converting Junctions to Network Topology

After finding all junctions including endpoints and throat junctions, the final geometric and topological features of the network can be obtained. **Figure 2h** shows the final set of junctions for the image of overlapping spheres. In this image, the throat junctions are the ones found using the maximum filter method. The junctions are labelled, indicated by their color in **Figure 2h**, and overlayed with the skeleton, which is shown in purple. Throats are found by removing all junction voxels from the skeleton, leaving a fragmented skeleton, which is also labelled. The labels in these two images correspond to the pore and throat index of the network. Generating the adjacency list (which pores are connected to which, and by which throat) is done by analyzing the image of labelled throats. Each throat is isolated, dilated by 1 pixel, then checked to see which pore junctions are overlapped. The adjacency list is generated by noting the two pore labels (e.g. i and j) on the row corresponding to the throat label being analyzed (e.g. k), thus row k of the adjacency list will contain $(i,\ j)$. The final adjacency list will be an $N_t x2$ array, where N_t is the total number of throats in the network. While finding throat connections, geometric properties of the throats are also found. The area is computed using the walker method, and the inscribed throat diameter is found as the minimum distance transform value along a throat. As discussed in Section 2.2, the majority of peaks in the distance transform do not lie on the skeleton. Therefore, it was found that MAGNET underestimated pore and throat sizes compared to SNOW, which determines these sizes from peaks in the distance transform. To remedy this problem, a maximum filter was applied to the distance transform, using a small spherical structuring element with a radius of 2 voxels, prior to finding the minimum throat radius.

Table 1: Pore and throat properties returned by MAGNET.

Pore Property	**Description**
Coordinates	The pore coordinates are an N_p – by – 3 array (N_p is the number of pores) of x, y, and z coordinates corresponding to the centre of the pore. In a cluster of junctions, the pore center is the voxel with the largest distance transform. The coordinates are multiplied by the resolution or length of a voxel.
Inscribed Diameter	This value is taken as two times the distance transform at the pore centre. It is the diameter of the pore that does not overlap with any adjacent grain voxels. This distance is multiplied by the voxel size to get pore diameter.
Throat Property	**Description**
Connections	The throat connections are a N_t – by – 2 array (N_t is the number of throats) where each row contains the indices of two pores that are connected to one another. The skeleton is used to determine the connections between pores.
Area	This is the area calculated using walkers as explained in section 2.5. By default, MAGNET sends out walkers only from the throat voxel with the minimum distance transform, but if more throat voxels are used, only the minimum throat area along each throat is returned.
Inscribed Diameter	This value is two times the minimum value of maximum filtered distance transform along the throat. It is the diameter of the throat that does not overlap any adjacent grain voxels. This distance is multiplied by the voxel size to get throat diameter.
Equivalent Diameter	This is the diameter of the circle with the same area as the throat cross-section. It uses the throat area property previously calculated by walkers in section 2.5. The result is multiplied by the voxel size to get a throat diameter.

After finding throat properties, pore properties are found including pore coordinates and pore size. This is accomplished in the same manner as for throats, by analyzing each labelled cluster of pore junctions. The pore coordinates correspond to the voxel location since these junctions were already reduced to a single voxel that lies on the skeleton and has the maximum distance transform value. The indices of the resulting pore center are multiplied by the image resolution or voxel size to get an x, y, and z coordinate that is returned as an $N_p x3$ array where N_p is the number of pores. For pore size, the inscribed diameter is the value of the distance transform at the pore center, also obtained after applying a maximum filter. **Table 1** provides a summary of all pore and throat properties returned by MAGNET.

3. SIMULATION AND VALIDATION

In this section, MAGNET is demonstrated on a 400^3 image of Berea sandstone obtained from the work by Dong and Blunt (6). The network extracted by MAGNET is compared to a network extraction by SNOW, lattice-Boltzmann method, and drainage by image-based sphere insertion. After which, the computational time of MAGNET is broken down by each image processing step and compared to SNOW.

3.1. Network Extraction of Berea Sandstone

Prior to network extraction, all blind pores were filled, and floating solids trimmed, yielding a porosity of 19.54%, compared to an initial value of 19.65%. **Figure 8** shows the resulting network extractions comparing SNOW and MAGNET, overlaid with the voxel image. Excellent fitting of pores inside and filling of the porous regions is demonstrated by MAGNET, and the two network extractions appear to compare reasonably well in terms of topology and pore size.

Visual inspection of the two approaches looks as though the SNOW extraction has fewer pores but higher connectivity than the MAGNET extraction. In general, MAGNET has more pores and throats than SNOW with 17,968 pores and 24,373 throats for MAGNET compared to 12,374 pores and 21,696 throats for SNOW. The average coordination number for MAGNET is less than the average coordination number for SNOW with 2.71 for MAGNET and 3.51 for SNOW. This can be attributed to the fact that MAGNET keeps nearby pores and allows them to overlap, while SNOW is more aggressive with merging nearby pores. The total network extraction took 55 seconds for MAGNET (without parallelization) and 3 minutes and 21 seconds for SNOW on a laptop with a 2.4 GHz Intel Core i5-1135G7 processor and 12 GB of RAM. The time that it took MAGNET to calculate the throat area was 12.8 seconds using 10 walkers per throat.

The pore and throat size distributions for the inscribed diameters determined from MAGNET and SNOW networks taken from the Berea sandstone are compared in **Figure 9**. The distributions for both pore and throat sizes are in excellent agreement. The average pore diameter is 32.5 μm for SNOW and 33.1 μm for MAGNET, while the average throat diameter is 24.8 μm for SNOW and 26.5 μm for MAGNET. One difference is that MAGNET has more smaller pores. Here, MAGNET has about 25% more pores with a diameter less than 20 μm, and in other cases it has been observed that MAGNET has twice as many small pores. One possible explanation for the number of small pores could be the dead-end branches in the skeleton that were not trimmed away. In the past, other medial axis extractions have trimmed away these dead-end pores since these pores do not contribute to flow (18). However, in simulations with reactions occurring at the interphase, such as battery discharge, these pores do contribute to the performance and thus should not be ignored. It would also be possible to increase the threshold at which junctions

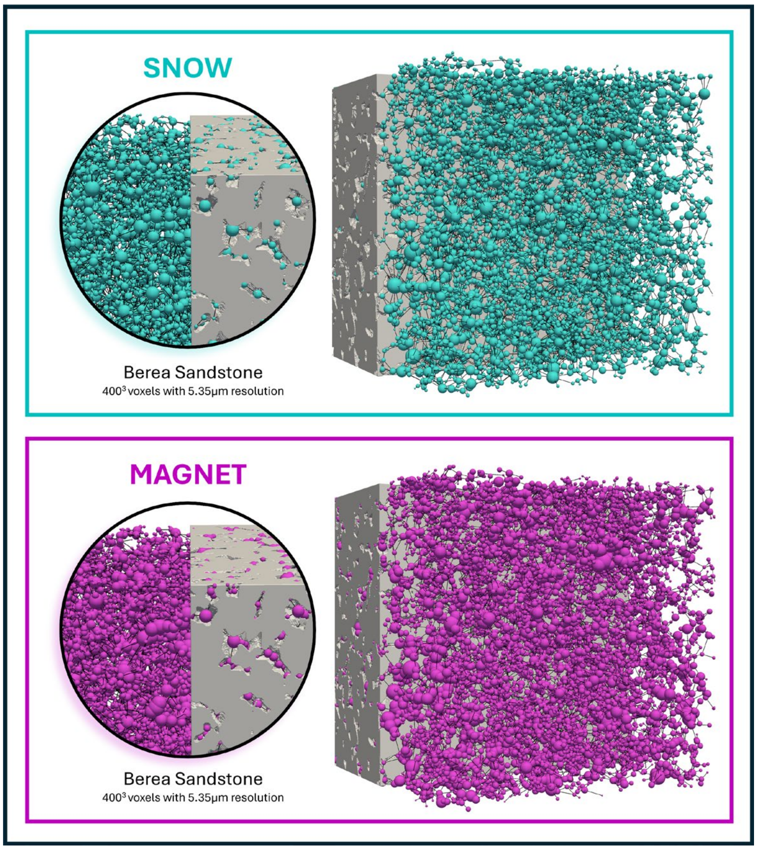

SNOW
Berea Sandstone
400^3 voxels with 5.35μm resolution
MAGNET
Berea Sandstone
400^3 voxels with 5.35μm resolution

Figure 8: A visualization of network extractions for 400^3 image of Berea sandstone with a 5.35μm resolution comparing SNOW and MAGNET extractions.

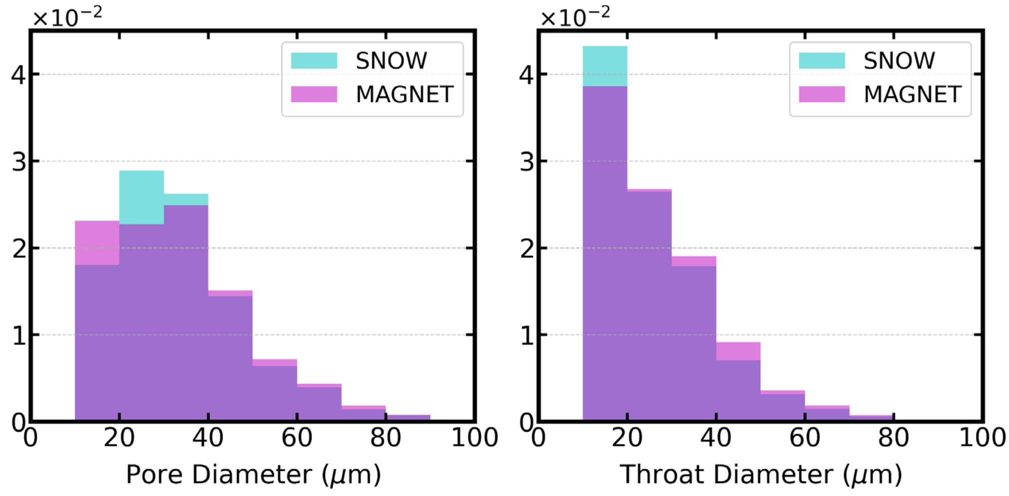


Figure 9: Pore and throat size distributions for SNOW (blue) and MAGNET (purple) extractions. The inscribed diameters are used for comparison.

are merged into a single pore, which would decrease the number of pores in the network and shift the size distribution to larger pores.

3.2. Validation of Network Extraction

3.2.1. Capillary Pressure Curves

A mercury intrusion porosimetry simulation was performed on both MAGNET and SNOW network extractions to compare capillary pressure curves. An image-based drainage simulation was also performed and considered as a ground truth. This method used a binary erosion followed by dilation using progressively smaller spherical structuring elements to find the invading fluid configuration. Image-based drainage is known to overestimate the intrusion capillary pressure since invading fluid must overcome the pressure corresponding to the inscribed diameter. Consequently, the inscribed diameter was used in both SNOW and MAGNET calculations. Meanwhile, `OpenPNM` was used to run a drainage simulation on the two extracted networks. The drainage simulation works by using the Young-Laplace equation to calculate the throat entry pressure from the throat diameter. **Equation 3** is the Young-Laplace equation where D_i is the diameter of throat i, γ is the surface tension, and θ is the contact angle of the invading fluid. Mercury with a surface tension of 0.4791 N/m and contact angle of 140 degrees was used as the invading fluid. All simulations were set to have mercury invade from all sides.

$$P_c = \frac{4\gamma}{D_i}\cos(\theta) \tag{3}$$

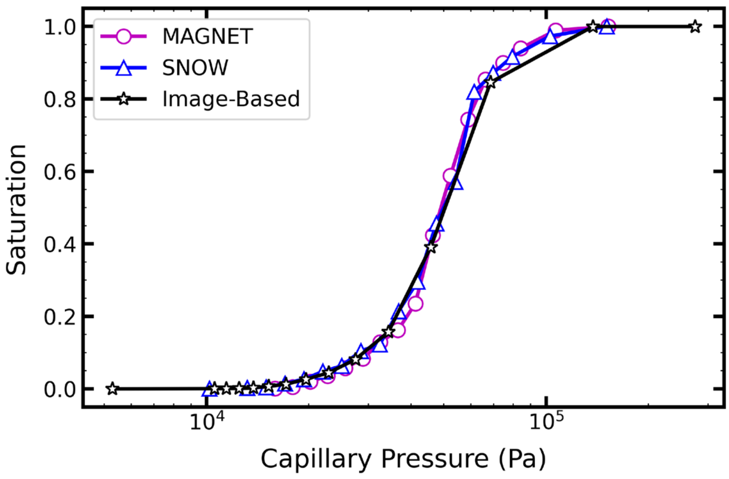


Figure 10: Resulting porosimetry curves from SNOW (blue) and MAGNET (purple) extractions on the Berea sandstone sample. Mercury was used as the invading fluid with a surface tension of 0.48 N/m and a contact angle of 140 degrees. The black curve shows the results from an image-based simulation that was used as the ground truth in comparison to results from the two pore networks.

Figure 10 shows the resulting capillary pressure curves for SNOW (blue), MAGNET (purple), and the image-based (black) simulation, all using inscribed throat diameter for fair comparison. The results show that all three agree very well. It should be reiterated that applying a maximum filter to the distance transform was necessary in matching drainage curves, which reinforces the idea that the skeleton does not follow maximums perfectly. This finding suggests that a new algorithm for obtaining a skeleton that follows the maximums in the distance transform could be a useful avenue of research, but this is outside the scope of the present work.

3.2.2. Permeability Coefficient

After extracting the network using MAGNET, a flow simulation was written and run to estimate the permeability of the Berea sandstone and results were compared to SNOW and lattice Boltzmann simulations on the same image. The open-source Python package `OpenPNM` (version 3.1.1) was used to carry out the simulation for MAGNET and SNOW networks (32), and the permeability values taken from Yi et al. (38) were used for comparison to lattice Boltzmann. In the pore network simulations, the hydraulic conductance was calculated using the cone and cylinder geometry models already available in `OpenPNM`. The hydraulic conductance, g_i^h, of arbitrary shape i, was calculated using **Equation 4** where λ_i^h is the hydraulic size factor and μ is the viscosity of the working fluid, which is assumed to be water or 1 cP.

$$g_i^h = \frac{\lambda_i^h}{\mu} \tag{4}$$

Equation 5 shows how to calculate the hydraulic size factor, λ_i^h, assuming negligeable inertial loss for a shape of varying cross-sectional area, $A_i(x)$, along length, l_i, of throat i (1). The specific polar moment of inertia is given by I_p^* which is calculated by $\frac{1}{A^2}\int y^2 + z^2 dA$, and depends on the shape of the cross-section. The specific polar moment of inertia, for a circular cross-section as is the case for cones and cylinders, is ½π.

$$1/_{\lambda_i^h} = 16\pi^2 \int_0^{l_i} \frac{I_p^*}{A_i(x)^2} dx \tag{5}$$

Assuming a resistor in series like model, the overall conductance for a pore-throat-pore conduit can be calculated from **Equation 6**. G_{ij}^h is the hydraulic conductance for a pore-throat-pore conduit connecting pore i and pore j.

$$G_{ij}^h = \frac{1}{1/_{g_i^h} + 1/_{g_{ij}^h} + 1/_{g_j^h}} \tag{6}$$

Finally, after defining the hydraulic conductance, flow was simulated on the Berea sandstone image. The boundary pores were labelled in all three directional axes and flow simulations were performed along each direction. The permeability, K, was obtained along each axis and then averaged to estimate the permeability of the sample. The permeability was calculated from Darcy's law (**Eq. 7**) where Q is the

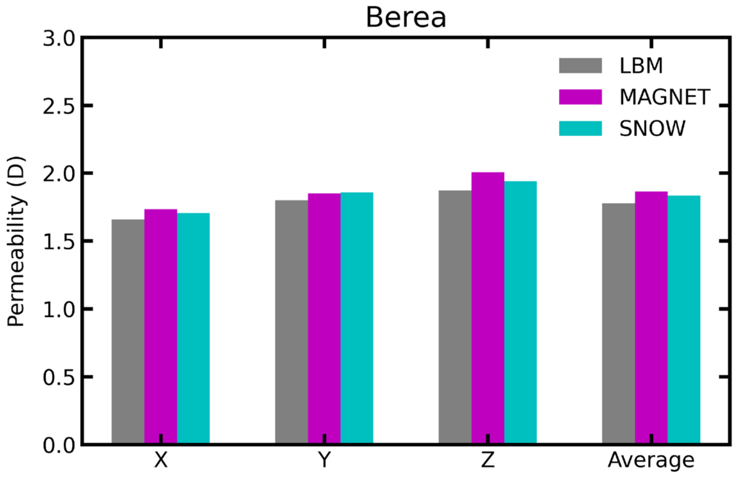


Figure 11: Predicted perme-ability of the $\mathbf{400^3}$ voxel image of the Berea sand-stone with a resolution of 5.35 µm are plotted and compared for lattice Boltzmann (grey), MAGNET (purple) and SNOW (blue) simulations. The perme-ability measured in each direction x, y and z are plotted along with the average of all three directions.

steady state flow rate estimated going through the rock sample. The length of the sample is denoted by L and the cross-sectional area perpendicular to flow is denoted using A.

$$K = \frac{Q\mu L}{A\Delta P} \tag{7}$$

The resulting permeability values from each case are shown in **Figure 11**. The permeabilities, averaged for all three directions, were 1.86, 1.84, and 1.78 darcy for MAGNET, SNOW, and LBM respectively. Note that the LBM estimate of permeability was retrieved from the work by Yi et al. (38) who performed lattice-Boltzmann on the exact same Berea image. Compared to LBM, MAGNET had an error of just 4.9%. To estimate permeability, the equivalent throat diameter was used. This corresponds to the diameter of a circle with the same area as the throat, as computed by the walker method outlined above. How the throat size is calculated highly influences the results from the permeability prediction. For instance, when using the inscribed diameter, the permeability predicted by MAGNET was 0.46 darcy, so 75% lower than the LBM result. This clearly highlights the value of the walker method for finding the throat area.

3.3. Computational Efficiency

To study the computational efficiency of MAGNET, the time taken was measured for each step of the process. **Figure 12** shows the resulting time breakdown for a 1000^3 image of packed spheres with a uniform radius of 10 voxels and 50% porosity. The total time to run the MAGNET extraction, prior to parallelization, and using the maximum filter to find throat junctions, was 35 minutes and 15 seconds. Compare this to the SNOW extraction which took 74 minutes and 40 seconds for the same image resulting in a speed up of over 2X. The operations which required the least amount of time were computing the distance transform and identifying the junctions/endpoints. The distance transform required only 54.9 seconds, making it the fastest operation and accounting for just 2.6% of the total extraction time. The next fastest operation was the finding of junctions and endpoints, which took 2 minutes and 46 seconds, which represents 7.9% of the total time. Finding throat junctions, however, took the most computational effort with a time of 11 minutes and 21 seconds or 32.2% of the total extraction time. Subsequently, the throat profile method was tried, reducing the total extraction time to 27 minutes and 28 seconds. This improvement was achieved by decreasing the time required to find throat junctions

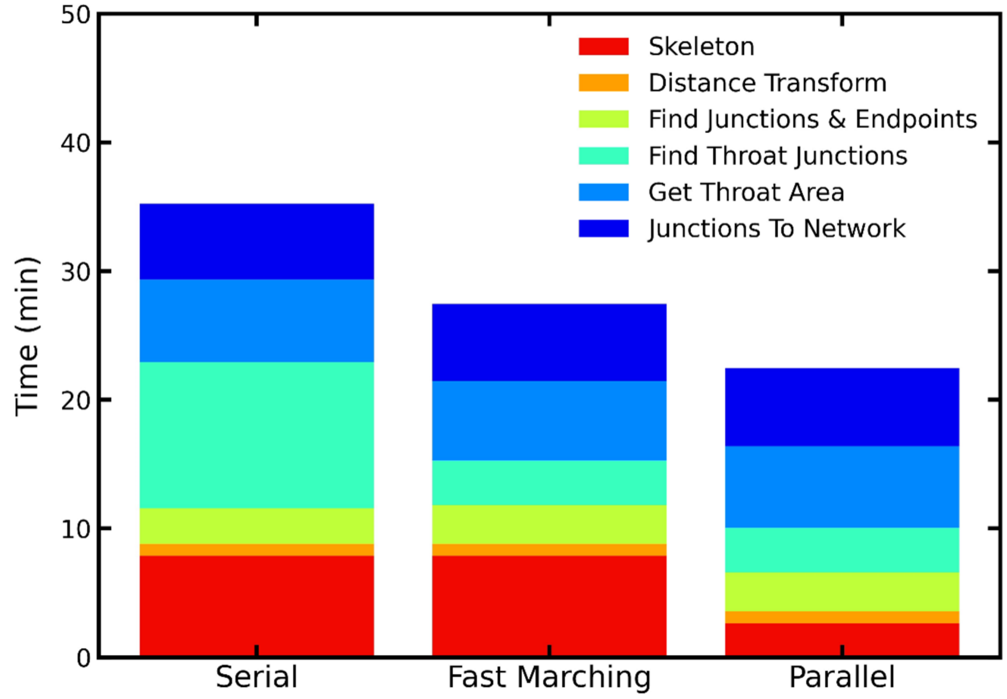


Figure 12: Time breakdown of MAGNET algorithm for a 1000^3 image of packed spheres with 50% porosity. The effect of using fast marching to find throat junctions and parallelizing the skeleton on the total extraction time is observed.

to just 3 minutes and 26 seconds, corresponding to only 12.5% of the total extraction time. The next most time-consuming step was the skeletonization, which required 7 minutes and 53 seconds or 28.7% of the total extraction time. To improve the extraction time, parallelization of the skeletonization was tried by dividing the image into 8 equal blocks with an overlap of 52 voxels (i.e. two times the maximum distance transform) for parallelization. The resulting time it took to compute the skeleton was reduced to just 2 minutes and 39 seconds or 11.8% of the total extraction time. Now, after using the throat profile method and parallelizing the skeletonization, the total extraction time is only 22 minutes and 28 seconds, and the resulting speed up compared to SNOW is 3.3 times for the same 1000^3 image. These tests for computational effort were carried out on a workstation with 250 GB of ram, 16 logical processors, and a 2.1 GHz Intel Xeon Silver 4110 processor.

After using the throat profile and parallelizing the skeletonization, the slowest step of the network extraction was finding the throat area. Calculating the throat area took 6 minutes and 21 seconds or 28.3% of the total extraction time, following parallelization. This time includes a gaussian filter that is used to help find the voxel with the minimum distance transform value along each throat as this voxel is considered a good candidate to calculate the throat area. A more detailed look at the time it takes to find the throat area using walkers revealed that only 44 seconds was spent by the walkers but an overwhelming 5 minutes and 37 seconds was spent just locating where to calculate the minimum. In this example, 359,485 throats are found, and 10 walkers are used per throat resulting in a total of approximately 3.6 million walkers. Extrapolating from **Figure 7c**, the predicted time for these many walkers, using adaptive stepping, is about 50 seconds, which is roughly what was measured here with a time of 44 seconds to complete walking. This is fast and demonstrates the benefit of using the distance transformation to adaptively control the step size walkers take. The most time-consuming part of finding the throat area is not walking, but finding the throat voxel to walk from which took 88.4% of the total time. The gaussian filter was removed in an effort to reduce this time; however, the permeability for the Berea sandstone unfortunately increased by 12.4% compared to the previous estimate. This is likely an effect of walkers just missing the grain boundary and marching beyond the throat constriction. Therefore, it is recommended to find and eliminate outliers in the distances walkers travel as part of future work.

Finally, the speed up of MAGNET was tested and compared to SNOW. The speed up was tested on artificially generated images of overlapping spheres with radius of 10 voxels and a porosity of 50%.

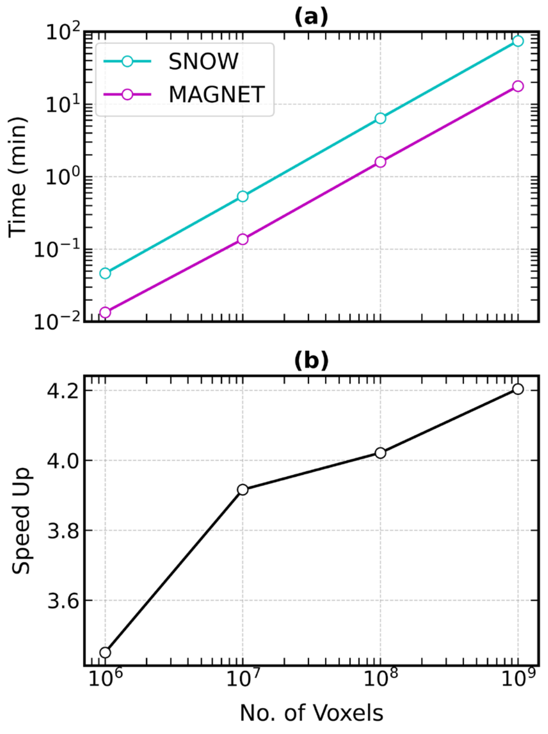


Figure 13: Network extractions were performed on different sized images using SNOW and MAGNET and the extraction times measured. Plot **(a)** compares the extractions times of SNOW to MAGNET using the serial skeleton. Plot **(b)** records the speed up is in the range of 3.4 to 4.2 for all images.

Floating solids were trimmed from the artificial images to create a realistic 3D image of packed spheres. Four different images with shapes of (100, 100, 100), (1000, 100, 100), (1000, 1000, 100), and (1000, 1000, 1000) were used. The MAGNET extraction was run in serial mode and did not calculate throat area or find throat junctions. **Figure 13a** compares the extraction time of MAGNET to SNOW in minutes for both SNOW and MAGNET at different image sizes. The extraction time recorded here for MAGNET includes the skeletonization. **Figure 13b** shows the speed up plotted against image size. The speed up using the serial skeletonization ranges from 3.4 to 4.2 compared to the parallel case which has a speed up in the range of 3.0 to 5.7. Parallelizing the skeletonization actually slowed down the extraction on the smallest image with 10^6 voxels because of overlap. Therefore, parallelizing is only recommended on larger images with at least 10^7 voxels.

4. CONCLUSIONS

The present work was built on previous medial axis-based network extraction tools by addressing several shortcomings or challenges with previous approaches. Firstly, a simple pore merging criteria was used that allowed pores to overlap, since this can be handled using appropriate pore-scale models in the pore network simulation stage. Secondly, two methods were presented for finding additional nodes along throat segments, which are important when considering reactive flow for instance (21). Thirdly, it became apparent that the skeleton does not always lie on the true peaks of the distance transform, meaning that inscribed diameters of pores and throats were being underreported. It was found that applying a simple maximum filter with a small round structuring to the distance transform largely remedied this problem by transposing the correct distance values onto the skeleton pixels. Lastly, we developed a novel method for finding the cross-sectional area of throats using walkers to probe the shape of the void space around each throat. Previous medial axis-based methods employed complicated and/or costly approaches to obtain this information, while the walker approach constituted just a small fraction of the total time for MAGNET.

Due to the relatively fast speed of the skeletonization algorithm, it was shown that MAGNET can be much faster than the widely used SNOW algorithm. And finally, it was also shown that skeletonization can be successfully performed in parallel using the "block-and-tackle" approach, provided the overlap between the blocks was sufficient to ensure the resultant skeleton was accurate. Using parallelization reduced the required time by more than 5X for large images.

To validate the extracted network, a 400^3 voxel image of a Berea sandstone sample with a resolution of 5.35 µm was analyzed by both MAGNET and SNOW. Strikingly similar pore and throat size distributions were observed except that MAGNET generally has more smaller pores (<20 µm). This was attributed to merging criteria, not trimming endpoints, and the skeleton missing some maximums on the distance transform. The versatile nature of the open-source network extraction means that users can easily change merging criteria or how endpoints are handled. Mercury intrusion was simulated on MAGNET and SNOW networks and excellent agreement with image-based drainage simulations was obtained when using the inscribed diameter. Finally, permeability was estimated using MAGNET and the average permeability for the Berea sandstone was within 5% error compared to a lattice Boltzmann simulation. The equivalent throat diameter, obtained as from the throat area computed using the walker method, was found to be essential for achieving such an excellent prediction of permeability.

In closing, MAGNET is a modernized medial axis approach to pore network extraction, that is both impressively fast and as accurate as existing tools. It is publicly available as part of the PoreSpy project (7), the source code for which is available on Github and the code is also deployed via the Python Package Index (PyPI) for effortless installation.

STATEMENTS AND DECLARATIONS

Author Contributions

Michael McKague: Methodology, Software, Validation, Data Curation, Visualization, Writing – Original Draft, Writing – Review & Editing. **Hamed Fathiannasab:** Methodology, Software. **Mohammad Amin Sadeghi:** Methodology, Software. Jeff Gostick: Supervision, Conceptualization, Resources, Funding Acquisition, Writing – Review & Editing.

Conflicts of Interest

There are no conflicts of interest to declare.

Data, Code & Protocol Availability

The network extraction tool MAGNET has been added to the open-source PoreSpy library and is available at https://github.com/PMEAL/porespy.

Funding Received

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)

ORCID IDS

Michael McKague	https://orcid.org/0000-0002-2792-5893
Hamed Fathiannasab	https://orcid.org/0000-0002-2416-139X
Mohammad Amin Sadeghi	https://orcid.org/0000-0002-6756-9117
Jeff Gostick	https://orcid.org/0000-0001-7736-7124

REFERENCES

1. Akbari, M., Sinton, D., & Bahrami, M. (2011). Viscous flow in variable cross-section microchannels of arbitrary shapes. *International Journal of Heat and Mass Transfer*, 54(17–18), 3970–3978. https://doi.org/10.1016/j.ijheatmasstransfer.2011.04.028
2. Al-Raoush, R. I., & Willson, C. S. (2005). Extraction of physically realistic pore network properties from three-dimensional synchrotron X-ray microtomography images of unconsolidated porous media systems. *Journal of Hydrology*, 300(1–4), 44–64. https://doi.org/10.1016/j.jhydrol.2004.05.005
3. Al-Raoush, R., Thompson, K., & Willson, C. S. (2003). Comparison of network generation techniques for unconsolidated porous media. *Soil Science Society of America Journal*, 67(6), 1687–1700. https://doi.org/10.2136/sssaj2003.1687
4. Bakke, S., & Øren, P.-E. (1997). 3-D pore-scale modelling of sandstones and flow simulations in the pore networks. *SPE Journal*, 2(02), 136–149. https://doi.org/10.2118/35479-PA
5. Delgado-Friedrichs, O., Robins, V., & Sheppard, A. (2015). Skeletonization and partitioning of digital images using discrete morse theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3), 654–666. https://doi.org/10.1109/TPAMI.2014.2346172
6. Dong, H., & Blunt, M. J. (2009). Pore-network extraction from micro-computerized-tomography images. *Physical Review E*, 80(3), 036307. https://doi.org/10.1103/PhysRevE.80.036307
7. Gostick, J., Khan, Z., Tranter, T., Kok, M., Agnaou, M., Sadeghi, M., & Jervis, R. (2019). PoreSpy: A python toolkit for quantitative analysis of porous media images. *Journal of Open Source Software*, 4(37), 1296. https://doi.org/10.21105/joss.01296
8. Gostick, J. T. (2017). Versatile and efficient pore network extraction method using marker-based watershed segmentation. *Physical Review E*, 96(2), 023307. https://doi.org/10.1103/PhysRevE.96.023307
9. Homann, H. (2007). Implementation of a 3D thinning algorithm. *The Insight Journal*. https://doi.org/10.54294/xjdr5f
10. Jiang, Z., Van Dijke, M. I. J., Sorbie, K. S., & Couples, G. D. (2013). Representation of multiscale heterogeneity via multiscale pore networks: Representation of Multiscale Heterogeneity. *Water Resources Research*, 49(9), 5437–5449. https://doi.org/10.1002/wrcr.20304
11. Jiang, Z., Wu, K., Couples, G., Van Dijke, M. I. J., Sorbie, K. S., & Ma, J. (2007). Efficient extraction of networks from three-dimensional porous media. *Water Resources Research*, 43(12), 2006WR005780. https://doi.org/10.1029/2006WR005780

12. Khan, Z. A., Elkamel, A., & Gostick, J. T. (2020). Efficient extraction of pore networks from massive tomograms via geometric domain decomposition. *Advances in Water Resources*, 14*5*, 103734. https://doi.org/10.1016/j.advwatres.2020.103734
13. Khan, Z. A., & Gostick, J. T. (2024). Enhancing pore network extraction performance via seed-based pore region growing segmentation. *Advances in Water Resources*, 183, 104591. https://doi.org/10.1016/j.advwatres.2023.104591
14. Lee, T. C., Kashyap, R. L., & Chu, C. N. (1994). Building skeleton models via 3-D medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6), 462–478. https://doi.org/10.1006/cgip.1994.1042
15. Liang, Z., Ioannidis, M. A., & Chatzis, I. (2000). Geometric and topological analysis of three-dimensional porous media: Pore space partitioning based on morphological skeletonization. *Journal of Colloid and Interface Science*, 221(1), 13–24. https://doi.org/10.1006/jcis.1999.6559
16. Lindquist, W.B. 1999. 3DMA General Users Manual. SUNY-Stony Brook technical report SUNYSB-AMS-99-20. Stony Brook, N.Y
17. Lindquist, W. B., Lee, S., Coker, D. A., Jones, K. W., & Spanne, P. (1996). Medial axis analysis of void structure in three-dimensional tomographic images of porous media. *Journal of Geophysical Research: Solid Earth*, 101(B4), 8297–8310. https://doi.org/10.1029/95JB03039
18. Lindquist, W. B., & Venkatarangan, A. (1999). Investigating 3D geometry of porous media from high resolution images. *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, 24(7), 593–599. https://doi.org/10.1016/S1464-1895(99)00085-X
19. Lindquist, W. B., Venkatarangan, A., Dunsmuir, J., & Wong, T. (2000). Pore and throat size distributions measured from synchrotron X-ray tomographic images of Fontainebleau sandstones. *Journal of Geophysical Research: Solid Earth*, 105(B9), 21509–21527. https://doi.org/10.1029/2000JB900208
20. Ma, C. M., & Sonka, M. (1996). A fully parallel 3D thinning algorithm and its applications. *Computer Vision and Image Understanding*, 64(3), 420–433. https://doi.org/10.1006/cviu.1996.0069
21. Misaghian, N., Sadeghi, M. A., Lee, K. M., Roberts, E. P. L., & Gostick, J. T. (2023). Utilizing pore network modeling for performance analysis of multi-layer electrodes in vanadium redox flow batteries. *Journal of The Electrochemical Society*, 170(7), 070520. https://doi.org/10.1149/1945-7111/ace554
22. Morigi, M. P., & Albertin, F. (2022). X-ray digital radiography and computed tomography. *Journal of Imaging*, 8(5), 119. https://doi.org/10.3390/jimaging8050119
23. Niblack, C. W., Capson, D. W., & Gibbons, P. B. (1990). Generating skeletons and centerlines from the medial axis transform. (1990) *Proceedings. 10th International Conference on Pattern Recognition*, I, 881–885. https://doi.org/10.1109/ICPR.1990.118234
24. Palàgyi, K., & Kuba, A. (1998). A 3D 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recognition Letters*, 19(7), 613–627. https://doi.org/10.1016/S0167-8655(98)00031-2
25. Prodanović, M., Lindquist, W. B., & Seright, R. S. (2006). Porous structure and fluid partitioning in polyethylene cores from 3D X-ray microtomographic imaging. *Journal of Colloid and Interface Science*, 298(1), 282–297. https://doi.org/10.1016/j.jcis.2005.11.053
26. Rabbani, A., & Babaei, M. (2019). Hybrid pore-network and lattice-Boltzmann permeability modelling accelerated by machine learning. *Advances in Water Resources*, 126, 116–128. https://doi.org/10.1016/j.advwatres.2019.02.012
27. Rabbani, A., Jamshidi, S., & Salehi, S. (2014). An automated simple algorithm for realistic pore network extraction from micro-tomography images. *Journal of Petroleum Science and Engineering*, 123, 164–171. https://doi.org/10.1016/j.petrol.2014.08.020
28. Schlüter, S., Sheppard, A., Brown, K., & Wildenschild, D. (2014). Image processing of multiphase images obtained via X-ray microtomography: A review. *Water Resources Research*, 50(4), 3615–3639. https://doi.org/10.1002/2014WR015256
29. Sheppard, A., Sok, R., Averdunk, H., Robins, V., & Ghous, A. (2006). Analysis of Rock Microstructure using High-resolution X-ray Tomography. In C. P. C. (Ed.), *Symposium Trondheim 2006 Proceedings* (Peer Reviewed ed., pp. 1-12). Article SCA2006-26 Society of Core Analysts. https://www.researchgate.net/publication/267551268_Analysis_of_rock_microstructure_using_high-resolution_X-ray_tomography
30. Shin, H., Lindquist, W. B., Sahagian, D. L., & Song, S.-R. (2005). Analysis of the vesicular structure of basalts. *Computers & Geosciences*, 31(4), 473–487. https://doi.org/10.1016/j.cageo.2004.10.013
31. Silin, D., & Patzek, T. (2006). Pore space morphology analysis using maximal inscribed spheres. *Physica A: Statistical Mechanics and Its Applications*, 371(2), 336–360. https://doi.org/10.1016/j.physa.2006.04.048
32. Silversmith, W., & Hilei, P. (2024). *Seung-lab/euclidean-distance-transform-3D: Zenodo release* (Version 2.4.0) (Computer software). Zenodo. https://doi.org/10.5281/zenodo.10815871

33. Takahashi, M. , Ahn, C. , and M. Kato. Hydraulic Stress Effect on 3 Dimensional Pore Network and Permeability Change. Paper presented at the 12th ISRM Congress, Beijing, China, October 2011. ISRM-12CONGRESS-2011-248. https://onepetro.org/isrmcongress/proceedings-abstract/CONGRESS12/CONGRESS12/ISRM-12CONGRESS-2011-248/100339
34. Thompson, K. E., Willson, C. S., White, C. D., Nyman, S., Bhattacharya, J. P., & Reed, A. H. (2008). Application of a new grain-based reconstruction algorithm to microtomography images for quantitative characterization and flow modeling. *SPE Journal*, 13(02), 164–176. https://doi.org/10.2118/95887-PA
35. Thovert, J. F., Salles, J., & Adler, P. M. (1993). Computerized characterization of the geometry of real porous media: Their discretization, analysis and interpretation. *Journal of Microscopy*, *170*(1), 65–79. https://doi.org/10.1111/j.1365-2818.1993.tb03324.x
36. Tranter, T. G., Kok, M. D. R., Lam, M., & Gostick, J. T. (2019). Pytrax: A simple and efficient random walk implementation for calculating the directional tortuosity of images. *SoftwareX*, 10, 100277. https://doi.org/10.1016/j.softx.2019.100277
37. Xiong, Q., Baychev, T. G., & Jivkov, A. P. (2016). Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport. *Journal of Contaminant Hydrology*, 192, 101–117. https://doi.org/10.1016/j.jconhyd.2016.07.002
38. Yi, Z., Lin, M., Jiang, W., Zhang, Z., Li, H., & Gao, J. (2017). Pore network extraction from pore space images of various porous media systems. *Water Resources Research*, 53(4), 3424–3445. https://doi.org/10.1002/2016WR019272
39. Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), 236–239. https://doi.org/10.1145/357994.358023