Pore space morphology analysis using maximal inscribed spheres

Dmitriy Silin\textsuperscript{a}, Tad Patzek\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a}Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 90R116, Berkeley, CA 94720, USA
\textsuperscript{b}University of California, Berkeley, 425 Davis Hall, Berkeley, CA 94720, USA

Received 12 December 2005; received in revised form 5 April 2006
Available online 16 May 2006

Abstract

A new robust algorithm analyzing the geometry and connectivity of the pore space of sedimentary rock is based on fundamental concepts of mathematical morphology. The algorithm distinguishes between the “pore bodies” and “pore throats,” and establishes their respective volumes and connectivity. The proposed algorithm also produces a stick-and-ball diagram of the rock pore space. The tests on a pack of equal spheres, for which the results are verifiable, confirm its stability. The impact of image resolution on the algorithm output is investigated on the images of computer-generated pore space.

One of distinctive features of our approach is that no image thinning is applied. Instead, the information about the skeleton is stored through the maximal inscribed balls or spheres (MIS) associated with each voxel. These maximal balls retain information about the entire pore space. Comparison with the results obtained by a thinning procedure preserving some topological properties of the pore space shows that our method produces more realistic estimates of the number and shapes of pore bodies and pore throats, and the pore coordination numbers.

The distribution of maximal inscribed spheres makes possible simulation of mercury injection and computation of the corresponding dimensionless capillary pressure curve. It turns out that the calculated capillary pressure curve is a robust descriptor of the pore space geometry and, in particular, can be used to determine the quality of computer-based rock reconstruction.

\textcopyright{} 2006 Published by Elsevier B.V.

Keywords: Pore space morphology; Connectivity; Two-phase flow; Capillary pressure

1. Introduction

Fluid transport in porous permeable rock is determined by the void space geometry and connectivity, and the solid surface/fluid chemistry. The ever-changing distribution of fluids in the pores of a gas- and oil-bearing rock must be understood to develop a successful hydrocarbon recovery or environmental remediation process. Although the length-scale of an oil field is measured in kilometers, the ultimate success of an oil and gas recovery scheme is the net result of countless displacement events at a scale measured in microns. The process-dependent redistribution of reservoir fluids during production and injection determines how much of the
initial hydrocarbons will be recovered and how much will be left trapped. Recent advances in micro-imaging of natural rocks [1,2], combined with advances in pore-level flow modeling [3,4], allow researchers and engineers to gain better insights into pore-level displacement mechanisms. In particular, credible predictions of the impact of the rock wettability and fluid properties on the relative permeabilities and capillary pressures, as well as on the trapped oil and gas saturations, are now possible [5–9].

A microscopic image of rock is a three-dimensional array of cubic atoms or voxels. Each voxel is assigned a non-zero value if it is attributed to the pore space and zero otherwise. A group of neighboring voxels can make a loosely-defined “pore throat” or a “pore body.” The pore throats control fluid flow, whereas the pore bodies provide fluid storage.

One of the difficulties in studying the microscopic structure of a porous medium is the absence of an elegant mean-field theory, such as the multiple-continua model. As a consequence, the microscopic rock models require storage and processing of huge amounts of data to characterize a tiny piece of rock. For example, a 5μm-resolution image of a cubic rock sample 2.5-mm on the side consists of 125 million voxels. In comparison, a reservoir simulation with an equal number of gridblocks would be a challenge even for a high performance parallel computer.

The number and size of the voxels in an image are limited by the resolution and viewing angle of the imaging device. The image itself is an interpretation of the reflection, absorption, attenuation, and diffraction patterns of electromagnetic waves. Each such interpretation is a solution of a series of inverse problems. The inversion errors are then combined with the uncertainty produced by segmentation, that is, assignments of the voxels, which are part solid and part void space. The representativeness of a digital image and the minimal requirements to the resolution are usually determined empirically.

In this context, attempts to develop efficient procedures of computer reconstruction of sedimentary rocks seem to be promising [11,10]. If such a procedure adequately reproduced a rock, it would be possible to create the corresponding digital image with theoretically arbitrarily high resolution. One has to bear in mind, though, that indefinite refinement of the resolution of the computer-generated rock images may be physically unrealistic.

The literature on processing images of natural rocks can be split into two periods. The earlier period is characterized by the development of basic concepts of mathematical morphology. A systematic presentation of early development and results of the theory of mathematical morphology was given in the monographs by Matheron in 1975 [12] and Serra in 1982 [13]. The concepts of skeleton and medial axis were inspired by early works of Motzkin [14] and Blum [15], and played a pivotal role in all subsequent investigations.

More recently, the revolutionary progress in imaging techniques [2,16–18] and computing power induced a new wave of morphological studies. Pore-network modeling made it possible to gain fundamental understanding of multiphase fluid flow in porous media. In conjunction with image analysis, it became possible to build models of porous media capable of predicting the fluid transport properties of rock, so important in oil industry [3,5,6,8,19–22]. Surveys of pore-network modeling are presented in Refs. [4,23–25].

Even if we concentrate purely on geometric issues and neglect errors introduced by image processing and interpretation, extraction of pore networks from the microscopic 3D images of rocks still poses challenging problems. Many “skeletonization” algorithms are based on thinning methods [17,18,26–32]. Thinning relies on removing the “redundant” elements of an image, while preserving certain topological properties of the entire pore space. However, intuitive extension of topological methods to digital images may be insufficient for rigorous analysis. The difficulties of porting the basic topological concepts [33], such as connectivity, Euler–Poincaré characteristic, etc., to discrete digital images are well known [34]. Tests of thinning algorithms on simple computer-generated images show that a refinement of the resolution can lead to less accurate results, which is an unwanted side effect. Simple illustrative examples demonstrate that the result of thinning may be unstable with respect to the choice of the starting point.

A characterization of the pore space geometry without application of thinning algorithms was proposed in Refs. [35–37], where some elements of the algorithms proposed here were developed. In particular, the characterization of skeleton as the set of centers of the maximal balls was employed. In Refs. [38–41], the capillary pressure curves were computed without pore network extraction. We also elaborate on this approach. A skeletonization method based on a complete catalog of shape primitives for 2D and 3D objects and suitable for higher dimensions was developed in Refs. [27,29,42,43].
In this paper, we focus on the analysis of pore space geometry. The image itself may have originated from computer tomography, a computer model of rock deposition, or any other source. We investigate how the results of image analysis are affected by image resolution and other variations of the input data. Our approach is based on the tools and concepts of mathematical morphology, as well as on an efficient implementation of the object-oriented algorithm design. Thus, all computer code used in this research has been written in C++ using standard template library (STL) [44] for data storage. The MIS method has been successfully applied to analysis of 3D geometry of the pore space of chalk [45]. The nanometer resolution images were obtained using focused ion beam (FIB) technology [2]. The capillary pressure curves obtained by MIS-calculations are in a good agreement with laboratory measurements.

This paper has the following structure. In the next section, we briefly overview the salient concepts of mathematical morphology. Then we describe how these concepts are implemented as objects in the algorithm. In the last section, we illustrate our approach with computations performed for digital images of both computer-generated and natural rocks. In Appendix, we briefly describe the cluster search and connectivity algorithms used in this study.

2. Basic concepts of mathematical morphology

The fundamental concepts of mathematical morphology, such as skeleton, medial axis, thinning, etc., are widely used in the literature on image processing. These concepts are usually illustrated in two dimensions where, for example, the medial axis of a channel-like structure is a curve passing through the points “in the middle” of that channel. Although even in 2D an intuitively clear definition can produce unexpected geometric features, in 3D the picture is much more complicated. It can be demonstrated that the skeletonization of a 3D channel, which is a graph of a set-valued mapping, can lead to extremely irregular structures [46,47]. Normally, in image-processing literature, it is tacitly assumed that in natural rocks such “exotic” structures are hardly possible and can be ignored. Following this tradition, we also assume that the constructions below are all legitimate and well defined.

To avoid ambiguity, we explain the basic mathematical morphology concepts used in this article. As a reference, we use the book by Serra [13], which apparently is the first systematic text on the subject. First, let us define the skeleton of a set of points. This concept is close but not equivalent to the more complex concept of medial axis. In general, the medial axis of a set is always a subset of the skeleton of this set. Each point in 3D space $\mathbb{R}^3$ can be characterized by three Cartesian coordinates $(x,y,z)$. These coordinates uniquely define a position-vector $r = (x,y,z)$. The Euclidean distance between two points is then defined as

$$\text{dist}(r_1, r_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.$$ 

A ball of radius $R$ centered at $r_0$ is a set of all points $r$ such that $\text{dist}(r, r_0) \leq R$. We will denote such a ball by $B_R(r_0)$. Let $M$ be a set in $\mathbb{R}^3$. In our applications, $M$ will be the pore space of a rock. The complementary set will be denoted by $M^c$. A ball $B_R(r_0)$ is called a maximal ball centered at $r_0$ if, first, it is a subset of $M$, and, second, it is not contained in any other ball, which is a subset of $M$. A point $r$ from $M$ is a point of the skeleton $S(M)$ if it is the center of a maximal ball. Clearly, for a given center $r_0 \in S(M)$, such a ball is unique. Any ball touching $M^c$ at two or more points is a maximal ball and, therefore, its center is an element of the skeleton. Centers of such maximal balls are also elements of the medial axis. In general, a ball can be maximal, but touch $M^c$ at a single point [13]. The center of such a maximal ball is an element of the skeleton, but not of the medial axis.

The definition of skeleton is dimension-independent, therefore a 2D version is straightforward. In the definition of the skeleton, we have chosen Euclidean metric due to its symmetry: the unit ball is invariant under an arbitrary rotation. Formally, any other metric can be used as well.

For simple geometric objects in 2D, the above-formulated definition describes structures, which resemble what one intuitively would call a skeleton, see Fig. 1. The skeleton of a more complicated 2D object can be bizarre. A rigorous application of the definition of skeleton results then in the structure shown in Fig. 2 as the dashed lines. The vertical “bones” of the skeleton correspond to the small perturbations of the upper boundary of the figure. Although these perturbations vanish to the right, the vertical bones are practically the
same. In 3D, even the skeleton of a simple convex body may have complex geometry. In Fig. 3, the skeleton of a rectangular parallelepiped makes a branching “film” cutting the pore space into six separate domains.

Methods of thinning were developed both to construct the skeleton and to eliminate the above-mentioned side-effects of skeleton generation, like the branches leading to the corners in Figs. 1 and 2. Thinning is usually based on the sequential applications of elementary operations in a hit-or-miss [13] transformation. This transformation finds a domain near the boundary of a set, which can be eliminated without altering the connectedness of the original set. Iterative elimination of the redundant domains found by this transformation thins the set and leads to a reduced structure representing, in some sense, the original set. The rigorous definition of hit-or-miss transformation [13] includes a pair of auxiliary sets. To obtain the desired result, this pair of sets must be chosen in an appropriate way. Due to their arbitrariness, the suitable choice of these sets is very difficult. The same pair, that successfully reduced the redundant side effects in one part of the set $M$, may create unwanted side effects elsewhere in the same set. Solved examples [13] of hit-or-miss transformations are mostly in 2D. Handling each case of an even simple irregularity requires custom approach. Development of a universal hit-or-miss transformation thinning algorithm routinely applicable to 3D digitized images of natural porous rock is hardly feasible.

Another group of thinning algorithms is based on the idea of sequential elimination of individual voxels, while preserving certain “topological” properties of the initial set $M$ at each iteration. The candidates for elimination are selected by some rule, but without a reference to the hit-or-miss transformation. Besides connectivity, these more sophisticated algorithms [30] also check values of topological invariants, e.g. Betti numbers. The elimination procedure is iterated until no further voxel can be removed, while preserving the values of these invariants. The remaining set of voxels is then declared as the medial axis. Seemingly, such an algorithm is based on the fundamental topological properties and should lead to robust sensible results. However, fundamental difficulties arise in finding appropriate discrete analogues of topological invariants. Originally, Betti numbers have been defined for differentiable manifolds with boundaries, see e.g. Ref. [48] and cannot be automatically applied to a cubic cut-off of the digitized rock pore space. So the way the discrete invariants are computed may not retain all the features of their continuous analogues. In addition, the recalculation of the discrete invariants at each iteration of the algorithm can be a daunting computational task.

The verifiability and reproducibility of the results of thinning can be problematic: a different orientation of the image, and multiple choices of the order in which the voxels are eliminated can alter the result significantly. As a simple illustrative example, let us thin a 2D set in Fig. 4, assuming that two pixels are connected if they...
Fig. 3. The structure of the skeleton of a simple 3D object, such as a rectangular parallelepiped, may not resemble what one intuitively calls a skeleton.

Fig. 4. The result of a thinning procedure strongly depends on the selection of the starting point: after removing pixel 3, the remaining set of pixels in figure (a) is complementary to the result in figure (b), although both are derived from the same original configuration.
have at least a common vertex and the pixel-removal procedure must preserve the connectivity. In other words, the criterion of pixel elimination is preserving the connection between the inlet and outlet voxels, 1 and 5. If we start with removing pixel 3, then pixels 2 or 4 cannot be removed without breaking the connectivity of the remaining set. Alternatively, sequential removal of pixels 2 and 4 leads to a set consisting of the pixel chain 1–3–5. Thus, applying the same principle and preserving connectivity, we obtain two mutually exclusive results, depending on the starting pixel: the interior parts of the two thinned sets complement each other. Although voxels 1 and 5 remain connected, the length and shape of the connecting chain are significantly altered.

This conclusion apparently holds true when the skeletonization is performed using shape primitives [43]. Using such primitives for measuring lengths of the skeleton elements will not lead to robust results because extraction of the skeleton by thinning is non-unique.

In a 3D image, there are at least three possible definitions of connected neighbor voxels. By the first definition, two voxels are connected if they have a common face. By the second one, every two connected voxels must have at least one common edge. Finally, by the third definition, two voxels are connected neighbors if they have a common vertex. According to the maximum number of neighbors of a voxel, the voxel connectivity is called 6-connectivity (first definition), 18-connectivity (second definition), and 26-connectivity (third definition). Two non-neighbor voxels are connected if they are end-points of a chain of voxels such that starting at one end, the other one can reached by a finite number of neighbor-to-neighbor steps. In 2D, 4 and 8-connectivity can be defined similarly. In the example in Fig. 4, the 8-connectivity has been used.

The definitions of pore bodies and pore throats are more complicated. Intuitively, in a fluid-bearing rock, pore bodies are the larger pore space openings where most of the fluids is stored. To a large extent, pore bodies determine the rock porosity. Pore throats are the narrow gateways that connect the pore bodies and determine the rock permeability. The coordination number of a pore body is the number of pore throats connecting it to other pore bodies. Although these concepts seem to be intuitively clear, their rigorous definition is not simple. Sometimes, pore bodies are defined as the junctions of three or more branches of the medial axis. As one can infer from the example in Fig. 3, even for a simple non-discretized 3D set such a definition makes little sense. In 2D, by this definition, a structure similar to that shown in Fig. 2 should have four distinct pore bodies with the coordination numbers equal to 3. In Section 4 pore bodies and pore throats are defined using maximal inscribed spheres. Although these definitions do not ultimately resolve all possible ambiguities, they are tested and verified on some geometric structures.

3. Voxel objects: Computing the skeleton

A discrete image is a set of voxels in space. Image processing algorithms operate on individual voxels to extract information about the entire image. Thus, a voxel object is the basic element of the pore space analysis. In addition to the three coordinates, its properties also include the radius $r_0$ of the maximal inscribed ball centered at the voxel. Depending on the task, this radius can later be complemented or replaced with the maximal radius $r_{\text{max}}$ of an inscribed ball covering this voxel, but not necessarily centered there. The voxel class can be extended to also include two lists of pointers to other voxels. These lists are conveniently used in the hierarchy definition below. In all calculations, the unit length is the linear size of one voxel. Therefore, the distances between voxels along each coordinate axis are integer numbers. Use of squared distances helps to avoid operations with floating-point numbers and the related round-off errors.

The first step of the algorithm consists of associating with each voxel the radius of the respective maximal ball. Starting from a zero-radius ball, i.e., the voxel itself, the radius of the ball is incremented by one until the ball hits a solid-phase voxel. A rough estimate of the complexity of such a search is equal to the total number of voxels in the pore space times the number of voxels in a maximal ball that can be inscribed in the pore space. This procedure can be computationally intensive for large images of high-porosity rocks. The algorithm complexity increases rapidly with refinement of the image resolution. The maximal radius search can be reduced by an incremental ball construction algorithm and using the information about the radii already computed for the neighbor voxels.
Once all voxel radii are computed, it is convenient to store all voxels objects in a list [44,49] sorted by the maximal radius. The reason is that often, in search operations, a computation applied to a voxel with a large maximal radius makes this operation unnecessary for many voxels with smaller radii. In addition, STL search algorithms are well debugged and documented.

In computations, only discrete balls are used. The discrete balls of small radii have shapes that only remotely resemble spheres, see Fig. 5. This circumstance brings uncertainty in computations at end-point saturation at high capillary pressure.

After the maximal radii have been assigned to all voxels, some of the respective balls may be subsets of the others. The included balls and respective voxels carry no additional information about the pore space and are redundant. Therefore, the second step in the algorithm is the removal of the included balls. To facilitate the search, an object called reference table is used. A reference table has the same dimensions as the image. An entry of this table with coordinates \((i, j, k)\) is a pointer to the pore voxel \(V\) centered at \((i, j, k)\) or a zero pointer if the voxel with coordinates \((i, j, k)\) is in the solid phase. To find all the maximal balls included in a given voxel centered at \(V_0 = (i_0, j_0, k_0)\), and with the maximal ball radius \(R_0\), it suffices to check all the voxels pointed by the reference table entries at \((i, j, k)\) such that

\[
\text{dist}(V, V_0)^2 \leq R_0^2.
\]

Once an included maximal ball is detected, the corresponding entry in the reference table is set to a zero pointer. After this operation is finished, the list of voxels is also updated by eliminating the centers of included balls.

It is interesting to note that in the “continuous world” a ball of radius \(R_1\) centered at \(c_1\) includes a ball of radius \(R_2\) centered at \(c_2\) if and only if

\[
R_2 + \text{dist}(c_1, c_2) \leq R_1.
\]

This criterion does not necessarily work for digital images: there are examples where the inclusion holds true, but the inequality fails. This paradox of discretization imposes some additional complexity on the procedure of elimination of included balls.

According to the definition of skeleton, the centers of the voxels, which remain on the sorted list after removal of the included maximal balls, constitute the discrete skeleton of the pore space. Clearly, the order, in which the radii are calculated, does not matter. Therefore, if a large image must be processed, it can be split into parts. Each part can be analyzed separately, and the results can be merged into the skeleton of the whole pore space. Therefore, a computer with a modest memory capacity can be used to process large images. Also,
the process can be parallelized, because the parts of the whole image can be processed independently of each other. Note that the parts of the entire image must overlap in order to eliminate the boundary effects.

A thinning algorithm is a one-way procedure: once a voxel is deleted, all information related to this voxel is removed for good. Therefore, it is impossible to undo these deletions and their cumulative effect can distort the result. In addition, the dependence of the result on which voxel should be removed first when there are multiple choices, makes an obstacle for processing an image by parts. In the proposed procedure, a voxel is removed only if the information about the entire pore space is preserved in the remaining list of maximal balls centers and radii.

4. Stick and ball diagram

Evaluation of the average coordination number is an important part of pore space analysis. Clearly, the result depends on how the pore bodies and pore throats are defined. As examples in Figs. 1 and 2 show, just the skeleton itself may carry very little information about the pore space structure. In this section, we introduce definitions of pore bodies and pore throats through the maximal inscribed balls and demonstrate on examples that they lead to a robust description of the pore space. In particular, the output of the proposed pore body—pore throat classification algorithm is independent of the starting point. The algorithm does not pursue the goal of complete pore-network extraction for single- or multi-phase flow simulations [3,5,6,21]. Rather, it robustly characterizes the pore space through a stick-and-ball representation.

4.1. Masters and slaves: Pore body and pore throat classification algorithm

To characterize the pore space, the redundant “ribs” corresponding to the corners, like in Figs. 1–3, and similar structures should be removed. Here we use the fact that a voxel object “knows” its maximal ball radius. Note that the maximal radius decreases along a rib leading into a corner. To formally characterize such a configuration, a hierarchy of the skeleton voxels needs to be established. We distinguish between master and slave voxels. If the maximal balls of two voxels overlap, the one with the larger radius is called a master and the other one is called a slave. Thus, the algorithm described above can be continued with the third step of search for the slaves of each voxel. Use of the reference table facilitates this task. First, for each current voxel a domain where potentially overlapping voxels can be located is determined from the radius of the maximal ball of the current voxel. Then, using the pointers stored for these locations in the reference table, the slave voxels are detected. After finishing this operation, each voxel object is appended with two lists of pointers to the masters and slaves of this voxel. For small radii, for which the balls have bizarre shapes (see Fig. 5), the hierarchy definition may need a slight modification, depending on the connectivity convention adopted in the analysis. Zero-radius voxels are declared slaves of all neighbor voxels with positive radii.

After computing this hierarchy, many voxels are masters and slaves at the same time. For example, Fig. 6 shows a part of rectangular pore space (the thick solid “C”), and the corresponding part of the skeleton (the dashed lines). It is a simplified 2D illustration, where voxels A, B, and C are shown as circles whose radii are the respective maximal radii. By definition, voxel B is a master of voxel C and a slave of voxel A. In this case, voxel A characterizes the size of the opening, whereas voxel B characterizes the narrowing into the corner, towards voxel C. Therefore, to characterize the pore body, it suffices to retain, as a master, only the voxel A with the largest maximal ball. Such a characterization is done through an enhanced hierarchy procedure. First, the master voxels, which are not slaves of other master voxels are selected. These “super” master voxels correspond to the local maxima of the maximal ball radii. In Fig. 6 such a voxel is A. Second, for all other voxels the lists of masters and slaves are rearranged: each “super” master voxels acquires as slaves not only its “immediate” slave, but the slaves of these slaves, etc. The principle “a slave of my slave is also my slave” is implemented. Using a depth-first type cluster search algorithm, see Appendix, each “super” master seeks all affiliated slaves. Reciprocally, all the slave voxels become affiliated with respective “super” master voxels only. In Fig. 6, voxel A will first find voxel B as an “immediate” slave, and then it will find voxel C as a slave of voxel B. Both voxels B and C become slave voxels of “super” master voxel A. Thus, after this search, voxel C is not a slave of voxel B and, respectively, voxel B is not a master of C. At the last step, the “intermediate” master voxels, like voxel B, are finally demoted to “pure” slaves and a new, smaller, list of master voxels is
created. In Fig. 6, after this operation only voxel A is a master voxel, and all other displayed voxels have become its slaves.

The procedure described in the previous paragraph results in the following picture. A list of master voxels, corresponding to the local maxima of the radii of maximal balls, is created. In a neighborhood of a voxel from this list, the maximal inscribed ball radii are equal to or less than the maximal ball radius at this voxel. If such a master voxel, say \( V^* \), is isolated, i.e., does not overlap with any other master voxel, then in all directions from the center of \( V^* \) the pore space narrows. Thus, an isolated master voxel can be associated with a pore body. The volume of the union of the maximal balls of all slaves for whom \( V^* \) is the only master, may be called the pore body volume. In Fig. 6, all the shown balls will be parts of the pore body associated with voxel A.

Sometimes, however, two master voxels may overlap. In such a case, the respective maximal radii must be equal, because otherwise the voxel with the smaller maximal radius would have been declared a slave of the other one by the procedure described above. In Fig. 6, if the pore space were continued to the right, voxel A would have a neighbor master voxel \( A' \) of the same radius. In such a case, it is natural to merge the respective master voxels into one pore body. Thus, more generally, a pore body is associated either with an isolated master voxel or with a cluster of master voxels in case of overlapping. The respective maximal ball (or one of the balls in case of multiplicity) defining the pore body will be a ball in the stick-and-ball diagram we are about to construct.

Now, let us check the slave voxels. Some of them, having only a single master, are parts of the respective pore bodies. However, there are slave voxels having two or more master voxels. Clearly, if a slave voxel, let us call it \( V_{s} \), has exactly two masters, then the pore bodies corresponding to these master voxels are connected via \( V_{s} \). For example, in Fig. 7, both voxels B and C are slaves of both voxels A and D. Hence, A and D are connected via B and C. It is natural to call the union of all balls associated with the voxels connecting two given master voxels a pore throat, which connects the respective pore bodies. In the stick-and-ball diagram, a pore throat connecting two pore bodies is depicted as a straight line segment connecting the centers of the balls associated with these pore bodies.

Although the above-introduced concepts are natural, they cannot eliminate all possible artifacts associated with the discretization of the pore space image. However, refinement of the resolution by reducing the voxel size reduces possible ambiguities.

4.1.1. Algorithm verification

To verify a physical model, numerical or analytical simulations should be compared with field or laboratory measurements. To verify a numerical algorithm, a test problem with a known analytical solution can be considered, and the numerical result can be compared with the exact one. Verification of an image analysis algorithm, like the one described above, is not as straightforward. On one hand, to make verification
representative, the test image must have sufficiently complicated geometry with internal openings and connections between them. On the other, the image should be small and simple so that the results of computations is transparent and verifiable. These two requirements are almost mutually exclusive. In addition, there is no single numerical criterion capable of evaluation of the quality of image analysis.

The empty space in a computer-generated packing of equal spheres seems to be a reasonable image analysis test. The spheres are packed in layers. Every layer where all spheres touch each other is sandwiched between two layers where each sphere is tangential to four spheres from one layer below and four spheres from one layer above Fig. 8a. The whole pack is shown in Fig. 8b, and its porosity, if the stencil 8a is applied indefinitely, is about 26%.

To make test more comprehensive and to make sure that the algorithm works regardless of the orientation of the image, the whole pack of spheres is rotated, as shown in Fig. 9. Then, a part of the image cut by a cube is analyzed. The pore space of this part is shown in Fig. 10. The constructed stick-and-ball diagram is displayed in Fig. 11.

Additional verification of the algorithm can be performed by analyzing the shapes of individual pore bodies and pore throats detected by the procedures described above. Inasmuch as these pore bodies and throats are only conventional terms introduced to characterize the geometry of the pore space, there is no clear boundary where a pore body ends and a pore throat begins. Thus, no quantitative criterion of whether the pore body and pore throat shapes are detected correctly can be applied, and the shape analysis is performed only visually. In Fig. 12a, two views of a pore throat are shown. Only the boundary voxels are displayed, so only “the walls” of the pore throat are seen. Nevertheless, the similarity with the channel formed by the three tangential spheres

---

Fig. 7. Both voxels B and C have voxels A and D as their masters. B and C can be called the pore throat connecting the pore bodies related to A and D.

Fig. 8. (a) A stencil of the sphere pack; (b) the whole sphere pack.
in Fig. 12b is clear. Similarly, Fig. 13a shows two projections of a pore body. Again, only the boundary voxels are displayed. The detected pore body structure can be compared with the shape of the void space surrounded by the six spheres shown in Fig. 13b. Such a combination of spheres is periodically repeated in Figs. 8b and 9.

The pore throats and pore bodies which are at the boundary of the image are cut and their shapes may be irregular. This fact has to be taken into account when the entire image is analyzed by parts and the partial results are later merged together: the parts into which the whole image is divided must overlap. The size of this overlap is determined by the sizes of grains and pores, and depends on the image resolution.

Since the sphere-pack in Fig. 9 is computer-generated, its image can be digitized at an arbitrarily high resolution. Therefore, the impact of the image resolution can be analyzed by comparing the results for the images of the same pack obtained at different resolutions. It is quite difficult to specify the rigorous requirements for a “sufficient” resolution of a rock image. However, numerical experiments with images of the
various computer-generated sphere-packs show that for an adequate description of the pore space, the resolution should be at least one order of magnitude finer than the representative sphere radius. Therefore, for example, the resolution of $5 \mu m$ may miss fine features of a rock with the grain size below $50 \mu m$. For imaging chalks or diatomites a super-high resolution imaging technique, such as the one proposed in Refs. [2,45], is needed.

Next, we present an example of application of the above-described algorithm to an image of Fontainebleau sandstone kindly provided to us by Schlumberger. The porosity of the sample is about 17%. From the whole image of $512 \times 512 \times 512$ voxels, a portion of the size of 200 voxels in each dimension was selected. The image of the pore space of the analyzed sample is shown in Fig. 14. First, the maximal balls of all voxels are found and displayed in Fig. 15. Note that these balls are shown as spherical objects of respective radii, not as discrete balls, examples of which are displayed in Fig. 5. For this reason, the flat surfaces cut by the boundary of the image, Fig. 14 have round shapes in Fig. 15. The result of the search for the master voxels is presented in Fig. 16. The displayed spheres have the radii of the maximal balls corresponding to the master voxels. Some of these balls may overlap. When the distance between two master voxels is small enough, the size of the throat connecting their associated maximal balls is comparable with the radius of the balls [59], and these voxels are
gathered in groups. There is a certain degree of freedom in specifying a criterion for grouping master voxels. In our simulations, we found that the distance between the centers not exceeding the radius could be such a criterion.

Our classification into “master–slave” voxels does not account for isolated voxels yet. There is a probability that such voxels are isolated due to the insufficient resolution of the image or the pore space may have disconnected cavities. Often, the isolated voxels are *singleton*, i.e. their maximal radii are zero. Fig. 17 shows the result of grouping and “clouds” of isolated singletons. Singletons can significantly distort further flow analysis. Hence, it is practical to clean up the image by removing singletons and small connected clusters, both solid and pore space, prior to any further computations [45].

Fig. 13. (a) Two views of a pore body. It is clearly seen that the pore body is an opening confined by six spheres in shown (b).

Fig. 14. The pore space of a $200 \times 200 \times 200$ image of Fontainebleau sandstone. The porosity is about 17%.
Finally, Fig. 18 shows the grouped master voxels, the singletons, and the links between each pair of the master voxels. We repeat that each master voxel stores detailed information about the respective pore body through its list of all slave voxels. Similarly, each link is presented according to the algorithm as a list of slave voxels having two masters. Such a list also provides detailed information about the actual geometry of the pore throat, cf. Figs. 12 and 13. The apparently weak connectivity of the sample is the result of its smallness. Only very few channels have been captured in Fig. 14.

5. MIS-calculation of the dimensionless capillary pressure

When the pore space is shared between two immiscible fluids in equilibrium, the wetting fluid occupies the corners of large pores and small pores, while the non-wetting fluid occupies the central parts of the pores it
invaded. The interface between these two fluids is a surface whose curvature is determined by the capillary pressure. Although in reality the fluid interfaces are not spherical, they can be approximated by spheres. The strongest deviation from the spherical shape of the interface is concentrated near the pore walls. The information about the maximal inscribed balls or spheres (MIS), obtained with the algorithm discussed above, can be used to reconstruct a capillary pressure curve. A procedure similar to the one described here was considered in Ref. [41] to analyze the pore space geometry of statistical reconstruction of pore space of sandstone.

A given capillary pressure, \( P_c = P_0 \), that is the difference between the pressures of the non-wetting and wetting fluid, determines the mean radius of curvature corresponding to an equilibrium interface at this capillary pressure level. For a sphere, the mean radius of curvature equals half of the radius of the sphere. The sphere radius corresponding to the selected \( P_0 \) will be called the threshold radius. It is natural to assume that at \( P_c = P_0 \), all the maximal balls whose radii exceed the current threshold radius cover the part of the pore space
that can be filled with the non-wetting fluid. The complementary part of the pore space remains filled with the wetting fluid. Calculation of the volumes occupied by each fluid relative to the total volume of the pore space yields the fluid saturations corresponding to a given capillary pressure. The described analysis is the essence of the maximal inscribed spheres method, MIS-calculations of dimensionless capillary pressure curve.

There are numerous reasons why a MIS-calculation is always approximate. With the bizarre shape of small “spheres”, cf. Fig. 12, one can expect higher degree of uncertainty for high capillary pressures. More accurate calculations at high capillary pressures require finer image resolutions. A high-resolution image may be technically unavailable or the size of such an image can require a huge volume of computations. Note that all calculations presented below are based on the assumption of capillary equilibrium and uniform wettability of the rock.

Storing voxels in lists sorted by the voxel maximal radii substantially simplifies MIS-calculations. For example, to evaluate the saturation, all the pore voxels covered by the respective balls are marked starting from the largest radius. Then, the evaluation of saturation reduces to counting the voxels marked by radii exceeding the threshold radius. This operation can be combined with various invasion-percolation algorithms to account for the history of fluid migration.

During fluid migration, whether it is drainage or imbibition or a sequential combination of the two, there is a possibility of development of enclaves of non-wetting fluid, which are entirely disconnected (trapped). This issue can be addressed by modeling an invasion-percolation process using breadth-first cluster search algorithms, see Appendix.

To investigate how the resolution affects the result, we apply MIS-calculations to the computer-generated images of a tilted pack of equal spheres, Figs. 9–10, which can be obtained at an arbitrarily high resolution. In Fig. 19, we present the dimensionless capillary pressure curves obtained at different resolutions. The abscissa is the wetting fluid saturation and the ordinate is the dimensionless capillary pressure, defined as the inverse curvature radius in voxel units. To calculate the physical capillary pressure, one has to rescale the computed values according to the actual resolution of the image and twice the coefficient of surface tension in the Young–Laplace equation. Such a rescaling was successfully applied to compute the capillary pressure curve for nanometer resolution images of chalk [45].

For the sphere pack, five dimensionless capillary pressure curves obtained at different resolutions are presented in Fig. 19. To make these curves comparable, an additional normalization has been applied, so that the curvatures are in the same physical length units for all curves. In the middle part of the saturation interval, the results are remarkably similar. However, at low saturations, the wetting fluid resides only in the narrow pores and in the corners of the large pores, and the results obtained at different resolutions differ significantly.

![Graph of dimensionless capillary pressure curves](image-url)

Fig. 19. These dimensionless capillary pressure curves vs. wetting fluid saturation were obtained from the images of the pore space of an equal-sphere pack (Fig. 14) at different resolutions.
At saturations close to unity, the non-wetting phase becomes disconnected, and can be trapped. Therefore, an experimental capillary pressure curve would not smoothly continue up to $S = 1$ as it is shown in Fig. 19, but instead will abruptly go to zero at a certain endpoint saturation $S = S^* < 1$.

After testing and verification, the procedure described above has been applied to analyze images of Fontainebleau sandstone. The images of four samples with average porosities between 12% and 21% were analyzed. Each image has dimensions of $512 \times 512 \times 512$ voxels and is split into parts. It is interesting to note that the variations of porosity between different parts within each sample are up to 100%. For each part, the dimensionless capillary pressure curve has been calculated, and the results have been compared to investigate stability, and, therefore, credibility of MIS-calculations.

Partitionings into cubes of sizes of 90, 180, and 270 voxels in each dimension have been considered. It turns out that the dimensionless capillary pressure curve is very stable with respect to the porosity variation: the capillary pressure curves obtained for the parts of different images are very close. In the first case, each image was presented as the union of $6 \times 6 \times 6 = 216$ subimages of $90 \times 90 \times 90$ voxels. The computed curves are shown in Fig. 20. The capillary pressure curves calculated for each part of a single image show some scatter, cf. Fig. 20(a)–(d). The most interesting observation is that the scatter within each part is comparable with the scatter among all 864 curves computed from all the parts and shown together in Fig. 20(e).

Fig. 20. These dimensionless capillary pressure curves vs. wetting fluid saturation were computed for parts of a Fontainebleau sandstone image. The parts have different average porosities: (a) $\phi = 12\%$; (b) $\phi = 13\%$; (c) $\phi = 17\%$; (d) $\phi = 21\%$. Plots a–d include each 216 curves, and (e) includes all 864 curves. In all plots the horizontal axis is the wetting fluid saturation and the vertical axis is the dimensionless capillary pressure.
In Fig. 21, the scattered circles have abscissae equal to the threshold wetting phase saturations, at which the non-wetting phase percolates, and the ordinates are the porosities of the respective parts. The 26-connectivity has been used in percolation analysis. The vertical and horizontal lines correspond to the median values of the porosities and the maximal saturations at which the wetting fluid is connected. The dashed line is the best linear fit. Note that the slope of this line is close to zero. Also, the scatter of the points in Fig. 21 is in sharp contrast with the almost coalescing capillary pressure curves in Fig. 20. Thus, at least at the scale and resolution of the analyzed images, the computed threshold saturation is practically independent of porosity.

If larger image parts are analyzed by the method described above, the dimensionless capillary pressure curves practically collapse together, whereas the scatter of the porosity-endpoint saturation computations does not change dramatically, see Fig. 22.

The dimensionless capillary pressure curves computed for different images may be different. Within each image, the curves computed for its different parts practically coincide with each other. Thus, the two groups of curves in Fig. 23 are clearly distinct. The lower group of curves corresponds to the image of average porosity 17%, whereas the upper group of curves is a combination of curves computed for both 12%- and 13%-porosity images. The curves for the 21%-porosity images are not displayed because they fall in-between these two groups. Such a non-monotonic behavior of the capillary pressure curves can be a consequence of image segmentation, under-representativeness of the image size, or the geometry of the rock pore space.

The fact that the computed dimensionless capillary pressure curves are almost the same for the samples with large variation of porosity suggests that it is the geometry of the pore space that determines the capillary pressure as a function of saturation. Note that saturation is a dimensionless parameter. Since the samples come from the same rock type, they must be of similar geometry and this similarity apparently makes the computed capillary pressures close to each other. The fact that the capillary pressure for a given pair of fluids is the product of the pore geometry and only weakly depends on the permeability can be observed from the experimental curves published a long time ago by Muskat [50]. A more significant impact on the fluids distribution and, therefore, on the capillary pressure curve, comes from the process, which has produced the given saturation. The capillary pressure curve is often expressed through dimensionless
Leverett’s $J$-function \[ P_c(S) = 2\gamma \sqrt{\frac{\phi}{k}} J(S), \] (1)

where $\gamma$ is the surface tension coefficient, $k$ is the absolute permeability of the rock, and $\phi$ is the rock porosity. The porosity and permeability of similar sedimentary rocks are usually correlated via a power function [52,53], see also Ref. [54]. The exponent of this power law can be different for different rocks, but is greater than one both in experiment and simulations. Substitution of such a correlation into Eq. (1) will result in a power-law
scaling for the capillary pressure with respect to either porosity or permeability. This scaling contradicts the observation that the computed capillary pressure curves for different fragments of the sample are only weakly affected by the variations of the porosity. This contradiction suggests that the influence of the pore space geometry on the capillary effects in two-phase flow needs more study.

Thus far, the capillary pressure curves have been computed without accounting for fluid migration. Such curves better model imbibition capillary pressure, because the wetting fluid is presumably connected through the pore corners and crevices at all times [60–62]. In primary drainage, the non-wetting fluid invades the pore space, which is initially filled with the wetting fluid. As the non-wetting fluid displaces the wetting one, it must remain connected to the inlet at all times. We assume that the flow is macroscopically one-dimensional, i.e. the fluids flow in and out only through two opposite faces of the cubic rock sample. To verify the results, three directions of flow were modeled for each image. Since no information about the orientation of the rock in its natural environment was available, we call these directions $X$, $Y$ and $Z$. The idea is as follows. We assume that the pressure of the wetting fluid inside the rock is constant, whereas the pressure of the non-wetting fluid near the inlet face increases. As the pressure increases, the non-wetting fluid first enters the largest pores and then continues its propagation [55]. As its invasion continues, the non-wetting fluid remains a single cluster connected to the inlet face. At a certain capillary pressure breakthrough occurs, and a sample-spanning cluster

Fig. 24. The rock porosity vs. wetting fluid saturation at breakthrough of non-wetting phase (a, c and e), and the dimensionless drainage capillary pressures vs. wetting fluid saturation (b, d and f) for 27 parts of the Fontainebleau sandstone image with the average porosity 17%. (a)-(b) drainage in the $X$-direction; (c)-(d) drainage in the $Y$-direction; (e)-(f) drainage in the $Z$-direction. In all plots, the horizontal axis is saturation. In the left plots, the vertical axis is dimensionless capillary pressure, whereas in plots (b)-(f) the vertical axis is porosity.
of pores containing the non-wetting fluid is formed. In simulations, a modification of a breadth-first cluster search algorithm described in Appendix was used.

In Fig. 24, we show the results for rock drainage simulated with invasion percolation. The entire image was partitioned into $3 \times 3 \times 3 = 27$ parts of 180 voxels in each direction. The results for the other images are similar and are not presented. It turns out that the flow direction does not significantly affect the results, i.e. the rock is isotropic. The mean breakthrough saturation is almost the same for the different parts and directions.

Thus far, we assumed the 26-connectivity among the voxels. In Fig. 25, we present the results of analogous simulations for the 18-connectivity. As one would expect intuitively, the scatter of the breakthrough saturations is higher than that for the 26-connectivity, but the difference is small.

The average wetting fluid saturation, at which the nonwetting phase percolates, increases with the image size. In larger images, the non-wetting phase spans the sample at lower capillary pressures. Clearly, the first spanning path of the non-wetting fluid increases in length with the size of the image. Therefore, the earlier percolation in a larger image is due to the fact that some connected, but tortuous, wide-pore paths can be cut in the smaller images.

Statistical methods for numerical reconstruction of natural rocks have been discussed in a large number of works, see e.g. Refs. [56–58]. For verification, a reliable criterion for comparison of the simulated and natural rock is needed. Usually, a two-point or other correlation function is used. In fact, a dimensionless capillary

![Graphs showing breakthrough saturations for 26- and 18-connectivity](image)
pressure curve also can be used to compare the images of natural and computer-generated rocks. Indeed, the results presented in this section demonstrate that the dimensionless capillary pressure curve does not depend on the porosity or the size of the sample, rather it depends on the intrinsic geometry and connectivity of the pore space.

6. Conclusions

We have presented a new robust approach to study the pore space morphology. Our approach is based on the fundamental concepts of mathematical morphology, summarized, e.g. in Ref. [13]. Based on the maximal radii analysis, we have introduced the concepts of pore body and pore throat. An efficient and stable algorithm distinguishing between the introduced pore bodies and pore throats has been developed and verified. This algorithm also leads to establishing the respective volumes and connectivity of the pore bodies and throats. Our algorithm has been tested on the images of computer-generated and natural sandstones. The algorithm tests on a pack of equal spheres, for which exact results can be verified visually, confirmed its stability. The impact of the resolution has been investigated using images of computer-generated sphere packs at different resolutions. The impact of inevitable artifacts of image discretization on the results of analysis decreases as the resolution of the image is refined.

The proposed algorithm produces a stick-and-ball diagram of the rock pore space. One of the distinctive features of our approach is that no image thinning is applied. Instead, the skeleton voxels are stored as objects including the radii of the associated maximal balls. In fact, these maximal balls retain information about the entire pore space. A comparison of our results with the results of a thinning procedure, which preserves some topological properties of the pore space [30], shows that our method produces more realistic estimates of the number of pore bodies and pore throats, and of the pore body coordination numbers.

The maximal inscribed spheres method, or, otherwise, MIS-calculations, produces a dimensionless drainage capillary pressure curve that simulates mercury injection. We conclude that the calculated capillary pressure curve is a robust descriptor of the pore space geometry, and it can be used to determine the quality of computer reconstruction of natural rocks. In addition, an appropriate scaling of this curve should predict the capillary pressure of the rock based on its 3D image. This scaling has been successfully applied for MIS-calculations of the capillary pressure curves for chalk. The details are reported elsewhere [45]. We have observed that the computed non-wetting phase breakthrough saturation in drainage practically does not depend on the sandstone porosity. Moreover, at least within the considered length scales, this breakthrough may occur at lower capillary pressures in larger sandstone samples. This circumstance is apparently associated with the fact that some connected wide-pore paths are tortuous and, therefore, may be cut by the image if the latter is not large enough.

Acknowledgments

This research was partially supported by the Assistant Secretary for Fossil Energy, Office of Natural Gas and Petroleum Technology, through the National Petroleum Technology office, Natural Gas and Oil Technology Partnership under US Department of Energy contract no. DE-AC03-76SF00098 to Lawrence Berkeley National Laboratory. Partial support was also provided by gifts from Chevron and ConocoPhillips to UC Oil, Berkeley. We are thankful to Schlumberger for providing the synchrotron images of Fontainebleau sandstone.

Appendix A. Cluster search and connectivity algorithms

In this section, we briefly describe two algorithms used in the MIS-calculations. These algorithms are based on the basic search principles [63] and are closely related to previously developed algorithms used on unstructured networks [64]. STL [44] was used to implement these algorithms in C++.

Cluster search is based on a depth-first type algorithm [63]. Assume that all pore voxels are stored in a list container. A list is a storage class, where insertion of an additional element or scanning the entire list is implemented through the mechanism of iterators [49]. In fact, it is convenient to keep the voxel objects
separately and operate with the list of pointers. To find the cluster containing a given voxel, the following procedure can be applied. A cluster is also a list where each voxel object has a property associating it with the cluster. This property can be an integer number, which is equal either to zero, if the voxel has not been associated with a cluster, or to the index of the cluster, with which this voxel has been associated. Let us denote this property \texttt{ClusterID}. Then, clusters search is just a recursive iteration of the following procedure:

\textbf{Function AddToCluster (ThisVoxel)}

\begin{verbatim}
for each neighbor pore voxel do
    if neighbor.ClusterID == 0 then
        neighbor.VoxClusterID = ThisVoxel.ClusterID
        add neighbor to the cluster ClusterID
        AddToCluster (neighbor)
    end if
end for
\end{verbatim}

Here \texttt{this.VoxClusterID} is the index of the cluster of the current voxel. The recursion consists in repeating this procedure for each newly assigned voxel. The definition of neighbor voxels depends on the adopted connectivity definition. If the cluster search is performed only among voxels filled with a given fluid phase, then only neighbors filled with the same fluid are considered.

The connectivity of a fluid phase can be investigated using a \textit{breadth-first} type algorithm [63]. The algorithms presented here verify whether there exists a fluid phase cluster spanning between the inlet and outlet faces and outputs the length on the shortest path. If there is no spanning cluster, then the procedure returns \(-1\). The algorithm operates with two lists of voxels. The first one, \texttt{CurrentList}, initially includes all inlet voxels filled with the fluid phase \(F\). The other one, \texttt{NextList}, is initially empty and is filled at each iteration. The above-introduced voxel property, \texttt{ClusterID}, can be used to mark whether the voxel has been already visited or not. At the beginning, we put \texttt{ClusterID} = 0 for all pore voxels. Then, the algorithms can be described in the following way:

\textbf{Require:} Inlet and Outlet faces are defined

\begin{verbatim}
MinPathLength = 1
CurrentList ← Inlet face voxels filled with fluid \(F\)
while CurrentList is not empty do
    for each voxel on Current list do
        for each neighbor pore voxel with fluid \(F\) do
            if neighbor is in outlet face then
                return MinPathLength
            end if
            if neighbor.clusterID == 0 then
                neighbor.ClusterID = 1
                add neighbor to NextList
            end if
        end for
    end for
    swap CurrentList and NextList
    clear NextList
    ++MinPathLength
end while
return -1
\end{verbatim}

As in the previous algorithm, the concept “neighbor” depends on the adopted definition of connectivity.

The complexity of the algorithms above is linear with respect to the number of voxels in the image. The number of times each voxel is visited by either algorithm is bounded above by the coordination number of the voxel, that is, the number of neighbors. The latter does not exceed the connectivity number.
A situation, where the porosity is equal to one, i.e., the whole image is pore space with no solid boundaries, is trivial with respect to the pore space geometry. However, for both algorithms, it is almost the worst case scenario with respect to the amount of computations.

References
