ARCHIVES OF MECHANICS
Arch. Mech. **74** (1), 39–64, 2022, DOI: 10.24423/aom.3861

An efficient method of tortuosity estimation

W. SOBIESKI

University of Warmia and Mazury in Olsztyn, The Faculty of Technical Sciences, 10-957 Olsztyn, M. Oczapowskiego 11, Poland, e-mail: wojciech.sobieski@uwm.edu.pl

The paper presents a comparative analysis of tortuosity calculations in two types of 2D random geometries: with non-overlapping circles and with overlapping squares. Both geometries were converted to binary geometries with different resolution. Next, simulations involving the Lattice Boltzmann Method were performed to obtain velocity fields in a pore space. Based on the obtained velocity fields, Hydraulic tortuosity and streamline tortuosity were calculated, based on the obtained velocity fields, for all considered cases. Hydraulic tortuosity was calculated with the methodology proposed by Koponen *et al.*, whereas streamline tortuosity was determined with the use of a new iterative algorithm. Two variants of the algorithm were proposed. Additionally, the obtained results were compared with selected formulas from the literature. The study demonstrated that calculations of streamlines exiting local inlet velocity maxima are a good alternative to calculations where all possible streamlines are taken into account. Computation time was significantly shorter and estimation quality was comparable.

Key words: porous media, tortuosity, streamlines, Lattice Boltzmann Method.

1. Introduction

Tortuosity, in addition to porosity or a specific surface, is one of the most important parameters characterising the spatial structure of a porous medium. Tortuosity was introduced by Kozeny [1] to correct the value of the hydraulic drop which occurs when a fluid flows through porous media. It was then applied by Carman [2] as a correction factor of filtration velocity. Tortuosity is defined as follows [3]:

$$\tau = \frac{L_p}{L_0}, \tag{1.1}$$

where: τ – tortuosity [–], L_p – the actual path length inside pore channels [m], L_0 – thickness of the porous medium [m].

In general, path length L_p may be defined as a geometrical quantity (geometric tortuosity) or a flow property (hydraulic tortuosity or streamline tortuosity – depending on the approach). Other kinds of tortuosity are also known (but

not considered in the paper), including diffusional, electric or acoustic tortuosity. It is important to note that pore space should form a connected network through which a fluid can flow. If this requirement is not met, parameter L_p cannot be calculated and loses its significance [4]. Tortuosity may be determined with the use of empirical formulas, calculations based on a discrete velocity field obtained for creeping flow, numerical algorithms which analyse pore channel geometry, or experimental data. The available methods have been reviewed in [5].

The available empirical formulas are simple to use, but the resulting tortuosity values are not very accurate. Our previous study [6] demonstrated that such formulas may produce significantly different results for the same data. Tortuosity ranged from 1.3654 to 1.6683 when calculated with empirical formulas designed for densely packed beds. The relative error between these results was almost 20%, which is quite significant. The comparison yielded 20 different empirical formulas, seven of which were designed for granular beds.

The second approach involves a discrete velocity field obtained for creeping flow. This field may be determined experimentally or with the use of numerical simulations. The Finite Volume Method (FVM), Lattice Boltzmann Method (LBM) and other methods may be applied in the latter case. The LBM approach is particularly useful for modelling complex systems due to the ease of operating on any complex geometry. If a velocity field is known, hydraulic tortuosity may be calculated with the following formula [7–11]

$$\tau = \frac{\langle |v| \rangle}{\langle v_x \rangle}, \tag{1.2}$$

where: $|v|$ – modulus of local flow velocity, v_x – X-component of local flow velocity (where X is the direction of the main flow). Triangular brackets denote the spatial average over a pore space. The described methodology, where the discrete velocity field was obtained from LBM simulations, was proposed by KOPONEN *et al* [10, 11]. In this approach, only one value of tortuosity is always known, and it is representative of the entire porous medium. The cited authors observed that "for a given obstacle configuration, the tortuosities calculated with different lattice resolutions were always found to be close to each other, and no systematic resolution effects were seen".

The same velocity field may be used to calculate streamline tortuosity [12]. In this approach, the lengths of individual streamlines must be first calculated. The average value of streamline tortuosity may be calculated as follows:

$$\tau = \frac{1}{n_s} \sum_{i=1}^{n_s} \frac{L_i}{L_0}, \tag{1.3}$$

where: n_s – number of streamlines calculated for a velocity field [–], L_i – length of the i-th streamline [m]. In addition to one representative result, many individual

values of streamline tortuosity can also be calculated, which is an advantage of this approach. The range of tortuosity values or their distribution may be useful when comparing two porous media with very similar average tortuosity.

In the context of the current article, it should be noted that details relating to streamline calculations, such as the required number of streamlines or the location of their origins, were not presented in [12]. The impact of lattice resolution on streamline tortuosity was not discussed either. The cited authors investigated velocity fields computed using the LBM in random pore systems that differed in porosity and consisted of overlapping circles (in 2D) or spheres (in 3D).

Koponen *et al.* and Matyka *et al.* have proposed empirical formulas for calculating tortuosity based on a porosity value. Another formula where tortuosity is a function of porosity and the normalised size of elements forming the rigid part of the solid body was proposed in [13]. The cited formulas are presented in Table 1. Symbol $|s|$ denotes the normalised size of a structural element [–] (size s divided by the number of grid nodes in the main direction of flow). In the context of the LBM, variable s denotes the size (length, width or the average value of these quantities) of a single obstacle in lattice nodes. This variable may be calculated directly when obstacles are rectangular in shape, or indirectly when obstacles are not rectangular in shape or when obstacles overlap. In such a case, the following formula may be applied:

$$s = \sqrt[n_{dim}]{\frac{n_1}{n_s}}, \tag{1.4}$$

where: n_1 – number of nodes marked as 1 in the geometry table [–] (details in subsection 2.2), n_s – number of structural elements (obstacles) in a porous structure [–], n_{dim} – number of the analysed dimensions (2 or 3).

Table 1. Formulas for calculating tortuosity in pore systems with known porosity ϕ.

	Source	Formula	Model constants
1	Koponen 1996 [10]	$\tau = 1 + 0.8(1-\phi)$	–
2	Koponen 1997 [11]	$\tau = 1 + a\frac{1-\phi}{(\phi-\phi_c)^m}$	$a = 0.65$, $m = 0.19$, $a = \phi_c = 0.33$
3	Matyka 2008 [14]	$\tau = 1 - pln(\phi)$	$p = 0.77 \pm 0.03$
4	Sobieski 2019 [13]	$\tau = \frac{a}{\phi^{b\|s\|^c}} + \frac{d}{\phi^{e\|s\|^f}}$	$a = 1.1$, $b = 0.33$, $c = -0.01$, $d = 5.3e08$, $e = 6.8$, $f = 2.86$

Tortuosity can be also calculated with the use of numerical algorithms that render the geometry of pore channels. These include the Path Tracking Method [15, 16] and the Connected Paths Algorithm [17]. The first method was designed to calculate geometrical tortuosity in granular beds composed of spherical or

quasi-spherical particles. The second method may be applied to two-dimensional binary images. Other methods based on the Random Walk technique [18] or the A-Star Algorithm [19] have also been proposed. The Path Searching Algorithm developed by the author in 2019 belongs to the same group of methods [13].

The main aim of the paper was to propose an approach for calculating streamline tortuosity where individual paths begin from local velocity maxima in the inlet plane. This approach reduces the number of paths needed to calculate representative tortuosity and decreases computation time. At the same time, the quality of tortuosity estimation does not decrease significantly. The search for faster methods of calculating tortuosity is important due to the emerging need to analyse increasingly complex systems. It should be emphasised that this study did not set out to investigate tortuosity as such. Such a goal would require an analysis of a much higher number of porous structures with different porosity, different obstacle shapes or obstacle size distributions.

The second aim of the present study was to compare the calculations of hydraulic and streamline tortuosity for selected two-dimensional random systems (similar to the systems used by Koponen *et al.* and Matyka *et al.*) and to test the impact of grid resolution on tortuosity values. In this context, the present study elaborates on the work done in [20], in which hydraulic tortuosity and geometrical tortuosity for similar random systems were compared.

The third aim is related to formula (4) in Table 1. This formula was proposed for two-dimensional random systems consisting of overlapping squares. Such systems as well as random systems composed of non-overlapping circles were also investigated in this study. The applicability of the above formula for analysing these types of porous media was also tested. In this context, the paper elaborates on the work done in [13].

All software used in this study was created by the author. For this reason, some information or solutions were shown in greater detail, in particular the self-developed algorithm for calculating streamlines and the method of determining local velocity extrema at the inlet.

2. Materials and methods

2.1. Geometry of pore structures

Two qualitatively different two-dimensional porous structures, i.e. circle-based and square-based, were used in this study. Both geometries are only examples of porous systems and do not directly represent real-world matter. The first system consists of randomly located and non-overlapping circles of different size. Since the random function produces values in the range of 0 to 1, it can be assumed that each direction of the domain has a unitary size (1 [m]). As a re-

sult, any two numbers drawn in this way can be directly treated as the X and Y coordinates of the centre of a circle. The diameter of each circle is also randomly determined based on an average diameter calculated as follows:

$$d_{ave} = 2\sqrt{\frac{A(1-\phi)}{\pi n_c}}, \tag{2.1}$$

where: d_{ave} – average diameter calculated based on the assumed domain size and porosity [m], A – area of the domain (in this case, equal to 1.0 [m^2]), ϕ – target porosity [–], n_c – number of circles in the domain [–]. The diameter of the i-th circle in the domain is calculated with the Monte Carlo method [21]:

$$d_i = d_{ave} 2\left[\frac{1}{n_d}\sum_{i=1}^{n_d} r_i\right], \tag{2.2}$$

where: d_i – diameter of the i-th circle [m], n_d – number of random values adopted in the model, r_i – i-th random number [–]. If n_d increases, the value in the brackets tends to be 0.5, and the standard deviation of circle size distribution decreases. Parameter n_d was set to 25 to obtain a set of circles with a somewhat different size.

It was also assumed that the circles cannot be grouped too closely. An additional parameter (σ, set at 0.025 [m] in the considered case) was introduced to determine the distance between circles during calculations. If the distance between a newly inserted circle and any already existing circle was smaller than σ, this circle was not considered and the drawing process was repeated. The number of circles in the investigated example was set at 50. The porosity calculated by the applied algorithm was equal to 0.6387. The average diameter and standard deviation of these particles were equal to 0.0941 [m] and 0.001873 [m], respectively. A certain disadvantage of the proposed algorithm is that it cannot be applied to generate systems with small porosity. Therefore, other algorithms, such as those based on the Discrete Element Method, should be applied to generate systems with low porosities [22, 23].

The second system consisted of partially overlapping squares with a fixed size. The algorithm was similar, but the distance between individual squares was not taken into account. As a result, the squares may have common parts and form complicated geometrical forms. The square had a side length of 0.05 [m]. Porosity was equal to 0.7995, and the system was composed of 97 squares. The two geometries are discussed in subsequent sections of this paper (see Fig. 6).

It should be mentioned that random generated structures may have consistent [10, 11, 14, 25] or inconsistent [8, 9, 24] geometry on opposite boundaries of the domain. In the paper the second, potentially more problematic, case is chosen.

However, it was not found that the lack of periodicity of the geometry caused any computational difficulties.

2.2. Lattice Boltzmann Method

The discrete Boltzmann equation is solved to simulate the flow of fluid in the Lattice Boltzmann Method. The Boltzmann equation may be saved as follows [26]:

$$\frac{\partial f}{\partial t} + \boldsymbol{v}\nabla_{\boldsymbol{x}} f + \frac{\boldsymbol{F}}{m}\nabla_{\boldsymbol{v}} f = \left(\frac{\partial f}{\partial t}\right)_{col} \quad , \tag{2.3}$$

where: $f(\boldsymbol{x}, \boldsymbol{v}, t)$ – the single-particle distribution function (where $\boldsymbol{x}$ is the coordinate and $\boldsymbol{v}$ is microscopic velocity), $\frac{\boldsymbol{F}}{m}$ — uniform external forces, $\left(\frac{\partial f}{\partial t}\right)_{col}$ – the collisional term. The function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ represents the probability (density) of finding a given particle at a given location in the space $\boldsymbol{x}$ and time t with a given velocity $\boldsymbol{v}$.

The LBM algorithm involves two main steps: a) a streaming process, and b) a collisional process. The streaming process may occur only between points that are regularly arranged in the space and form a structural lattice. In this type of lattice, gas can move in several directions which are represented by two numbers. The first number denotes the number of dimensions in the space (D2 or D3). The second number represents the number of directions in which lattice gas may move (e.g., Q27). The most popular LBM variants are D2Q9, D3Q15, D3Q19 and D3Q27. The BGK model [24] offers the simplest approach to defining the collisional term.

Equation (2.3) was implemented in the code in the following discrete form [27]:

$$f_i(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) - \frac{1}{\tau}[f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)], \tag{2.4}$$

where: $f_i^{eq}(\boldsymbol{x}, t)$ is the equilibrium function

$$f_i^{eq}(\boldsymbol{x}, t) = \rho w_i\left(1 + 3\boldsymbol{e}_i\boldsymbol{v}^{eq} + \frac{9}{2}(\boldsymbol{e}_i\boldsymbol{v}^{eq})^2 - \frac{3}{2}(\boldsymbol{v}^{eq})^2\right), \tag{2.5}$$

τ – relaxation time (the time required for the system to return to a state of equilibrium after being disturbed), $\boldsymbol{e}_i$ – the direction vector, ρ – lattice gas density, w_i – lattice weights (importance of a given direction). The right side of Eq. (2.4) represents the BGK collisional model. A gas movement in the main direction is forced by modifying the X-component of equilibrium velocity, which is accomplished by adding a term with the external mass force $\boldsymbol{F}$ [27]:

$$v_{i,x}^{eq} = v_{i,x} + \frac{F_x}{\rho}\tau. \tag{2.6}$$

If the values of the distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ in each i-th direction are known, macroscopic density and macroscopic velocity of lattice gas may be determined with the following formulas [27]:

$$\rho = \sum_{i=0}^{n_i} f_i \tag{2.7}$$

and

$$v = \frac{1}{\rho} \sum_{i=0}^{n_i} f_i e_i \tag{2.8}$$

where: n_i – a number of spatial directions in the model (equal 8 in the D2Q9 model; 0 is referred to the current lattice node).

Equations (2.7) and (2.8) are applied to calculate lattice gas density and lattice gas velocity. For the needs of this study, these values do not have to be expressed in real units because any scale factor introduced to Eq. (1.2) does not change the result. This approach was used by all the researchers mentioned in Table 1.

2.3. Geometry conversion

In this study, the Lattice Boltzmann Method was used to obtain a velocity field of lattice gas for creeping flow. However, to perform an LBM simulation, the previously developed geometries have to be converted to a regular grid (in this paper, the words "grid" and "lattice" are used interchangeably and do not have separate meanings) of nodes identified by successive integer numbers. The conversion concept is shown in Fig. 1. First, the real geometry must be covered by a regular grid of points. Next, all points are assigned a value of zero or one, depending on whether the point is located inside the pore part of the porous medium or not. The obtained matrix of zeroes and ones defines the geometry in

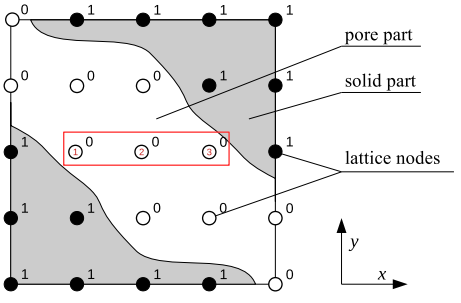

Fig. 1. The concept of converting a real geometry to a lattice grid.

the context of the LBM. The lattice for the circle-based geometry has a basic resolution of 200×200. Smaller grid resolutions were not considered because the width of individual pore channel cannot be less than 4 nodes [14]. In the analysed case, the minimum pore channel width was equal to 5 nodes. In general, grids with a coarser resolution can be included in the calculations, but a random pore structure would have to be generated with other parameters. In this case, the distance parameter σ in particular would have to be increased. The lattice for a square-based geometry has a basic resolution of 100×100. In this case, the width of pore channels may be limited to a single node due to the applied generation procedure. For this reason, such grids must be refined before LBM simulations are performed (refer to Subsection 3.1). It should be mentioned that after LBM simulations, every node with a zero value in the geometry matrix adopts a non-zero value of lattice gas velocity. In fact, the original matrix of zeros and ones does not have to be available at this stage. Geometry may be reconstructed on the basis of the velocity field.

Conversion a vector geometry to the binary geometry is always the lossy conversion. The original geometry cannot be accurately reproduced from a binary lattice of zeros and ones, especially in areas with curved surfaces, which are represented in it, in a stair-shaped form. Curved shapes also may be applied in LBM simulations, but more complicated boundary conditions at walls have to be implemented [28, 29].

2.4. Calculation of streamlines from a discrete velocity field

In the discussed method, a discrete velocity field is normalised at the beginning to obtain values between 0 and 1. For this purpose, every velocity component in the entire velocity field is divided by the maximum value of the velocity modulus. As a result, the maximum length of a velocity vector does not exceed the distance between two neighbouring nodes. It was assumed that the locations of grid nodes are identified by integer numbers. This grid may be defined as, for example, a grid of nodes in the LBM or as a grid of cell centres in the FVM or IBM. Only structural grids were used in this study, but other types of grids may be also applied, such as experimentally derived discrete velocity fields. The normalization process involves the following formulas:

$$\hat{v} = \frac{|v|}{|v_{max}|}, \quad \hat{v}_x = \frac{v_x}{|v_{max}|}, \quad \hat{v}_y = \frac{v_y}{|v_{max}|}, \tag{2.9}$$

where: $\hat{v}$ – normalised velocity modulus, $|v|$ – velocity modulus, $|v|_{max}$ – maximum value of the velocity modulus, $\hat{v}_{x,y}$ – normalised values of velocity components [–], $v_{x,y}$ – values of velocity components. Velocity may be expressed by physical units ([m/s]) or by lattice units (the approach used in this paper). The normalised variables are non-dimensional.

The locations (hereinafter referred to as Starting Points or SP) from which streamlines should begin and the number of streamlines are determined in the next step. Two cases were considered in this study. In the first case, streamlines began from every node in the first row of the grid in a direction (Y) perpendicular to the direction of the main flow (X). Only nodes belonging to pore space were considered. The maximum number of streamlines may be equal to grid resolution in the Y direction. In the second case, streamlines were calculated only from nodes for which the inlet velocity profile reached a local maximum. The following iterative procedures were developed for this purpose. First, the values of the velocity modulus for the first column of grid nodes in an additional indexed variable (one dimensional table) $v_{inlet}\langle 1 : n_y\rangle$ were remembered. Next, the locations (indexes) of all extrema in this table were searched and remembered. In the second step, two indexed variables were used: the previously mentioned v_{inlet} table and a new logical table (named $flag\langle 1 : n_y\rangle$). At the beginning, all elements of the logical table were set to FALSE. First, the location (index) of the maximum value in the v_{inlet} table was searched, and this value was set to zero. At the same time, the element in the logical table with the same index was set to TRUE. In the next iteration, the location of the maximum value in the v_{inlet} table was searched again. If the new location neighboured another extremum (identified based on the corresponding elements of the logical table), the logical value in the *flag* table was set to TRUE and a new search was initiated. Otherwise, the new location would be regarded as yet another extremum. In both cases, the velocity value of the current index was set to zero. The above procedure is very easy to implement, and its main advantage is that mathematical calculations are not required.

When a Starting Point (SP) was chosen, the velocity component in the main direction of flow was checked. If $v_x \geq 0$, then its location was saved as the first point of the current streamline (red point in Fig. 2). The location of the end of the normalised velocity vector was regarded as the second point of the same streamline (blue point in Fig. 2). To calculate the third path point (and every next one), velocity components in the last path point must be known. They may be calculated by interpolating the velocity field from the nodes forming a square (grey area in Fig. 2b) inside which the current path point is located. The appropriate weights must be taken into account because the influence of particular nodes depends on distance. The weights are defined as follows:

$$w_i = \frac{l_{sum}}{\hat{l}_i} \frac{1}{\frac{l_{sum}}{l_1} + \frac{l_{sum}}{l_2} + \frac{l_{sum}}{l_3} + \frac{l_{sum}}{l_4}}, \tag{2.10}$$

where: l_{1-4} – distances between the current path point and the neighbouring nodes [m], $l_{sum} = l_1 + l_2 + l_3 + l_4$ – sum of distances l_{1-4} [m].

The increase in the values of coordinates X and Y may be calculated as follows:

$$\begin{aligned} dx &= \frac{w_1\hat{v}_{x,1} + w_2\hat{v}_{x,2} + w_3\hat{v}_{x,3} + w_4\hat{v}_{x,4}}{4} \\ dy &= \frac{w_1\hat{v}_{y,1} + w_2\hat{v}_{y,2} + w_3\hat{v}_{y,3} + w_4\hat{v}_{y,4}}{4} \end{aligned} \tag{2.11}$$

where: $\hat{v}_{x,1-4}$ and $\hat{v}_{y,1-4}$ – components of normalised velocity in the neighbouring nodes [–]. In the adopted approach, it does not matter whether the points forming the current square belong to the pore space or to a solid body.

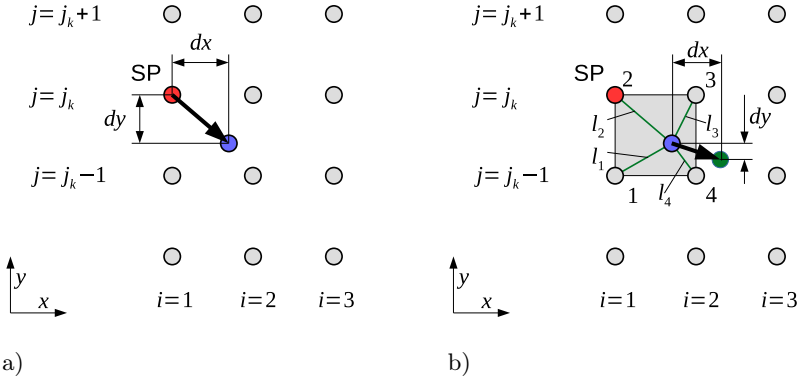

FIG. 2. Diagram of the algorithm for calculating streamline coordinates: a) in a Starting Point, b) in the second point of the path.

After the increase in the values of coordinates X and Y has been determined (2.11), the coordinates of the next streamline point (green point in Fig. 2b) can be calculated as follows:

$$\begin{cases} x_i = x_{i-1} + dx, \\ y_i = y_{i-1} + dy. \end{cases} \tag{2.12}$$

The operation of the proposed algorithm is presented on the example shown in Fig. 3. The streamline follows the velocity vectors. The streamlines do not cross each other, which confirms that the algorithm works correctly. The path length (L_p) can be calculated after the coordinates of all constituent points have been determined. Since the length of the domain in the main direction of flow (L_0) is also known (equal to $n_x - 1$), tortuosity can be calculated for every streamline and/or one average value representative of the entire velocity field. The distribution of the obtained data may be also analysed.

The tests revealed that the proposed algorithm had to be refined, in particular in the locations where velocity approached zero, including in the boundary layer.

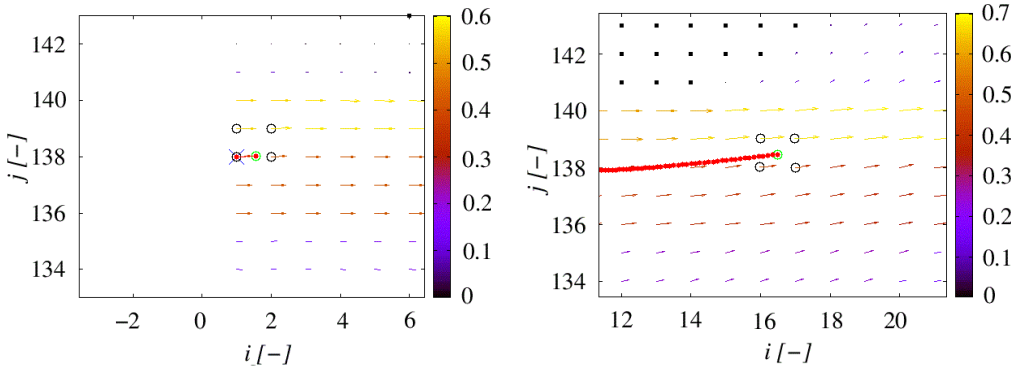

Fig. 3. Examples of algorithm operation: the first point of a streamline (left), and the i-th point of a streamline (right).

In this case, the path moves very slowly, and the direction of the streamline is usually lost due to the finite accuracy of a space discretization. It should be noted that in the LBM, lattice nodes are uniformly distributed in space, even close to the walls. This feature distinguishes the LBM from other approaches, such as the FVM, where different near-wall meshes are used as the standard solution to improve the quality of the results. In some cases, the path is directed towards the walls, which is impossible in real-world systems. If any path projects deeper into the imaginary line of the wall (dotted line in Fig. 4), the calculations are interrupted and the current path is not taken into account. Such cases are referred as a "dead end". In some cases, a path is not directed towards the wall, but reaches the stagnation area and becomes trapped. To eliminate such cases, the number of path points is calculated. The calculations are terminated if the number of path points exceeds the grid resolution in the main direction 100-fold. Such a case is referred to as "looping". Calculations are also interrupted if a streamline returns and ultimately exits the domain thorough the inlet. This exception is known as "reverse flow". Cases where the velocity component of the first point of the path is directed against the main direction of flow belong to the same group.

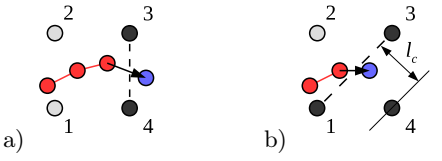

Fig. 4. Path penetration into: a) a flat wall, b) wall corner.

The periodic boundary conditions in the direction perpendicular to the main direction of flow are implemented in the developed algorithm. As a result, the

algorithm is consistent with LBM simulations (refer to Subsection 3.1). The following periodic conditions may apply:

$$\text{if } y_i \leq 1 \text{ then } y_i = n_y \tag{2.13}$$

and

$$\text{if } y_i \geq n_y \text{ then } y_i = 1, \tag{2.14}$$

where y_i denotes the Y coordinate of the current point of a streamline.

A different version of the algorithm had been previously tested. It was assumed that a bouncing procedure should be applied if a path attempted to penetrate a wall. The previously tested version of the algorithm produced highly similar results to those described in the present study, but computation time was several hundred times longer.

3. Results and discussion

3.1. Determination of discrete velocity fields

The LBM geometry obtained from the conversion procedure shown in Fig. 1 had to be refined to achieve the research objective. For this purpose, every individual node was replaced by a matrix of nodes with the same value (zero or one) and a size equal to $n_r \cdot n_r$ where n_r is the refinement factor. Five lattices were developed for each geometry. The refinement factor ranged from 1 to 5 for the circle-based geometry and from 4 to 8 for the square-based geometry. In the second case, the condition relating to the minimum number of lattice nodes was not fulfilled in the basic lattice. As a consequence, the refinement factor could not be less than 4. Higher values of the refinement factor were not taken into account due to the significant increase in computational power needed to perform LBM simulations. For example, in an input grid with a resolution of 200×200 and $n_r = 5$, the final number of nodes is equal to one million. All calculations were performed on a PC with an Intel Core 3.4 GHz processor, 32 GB RAM and xUbuntu 18.04 OS.

In the next stage, an in-house numerical code was used to simulate creeping flow through a pore structure with the use of the Lattice Boltzmann Method and the D2Q9 BGK model. The periodic boundary condition was applied to both directions. The no-slip conditions are specified at the walls. The external mass force (refer to Eq. (2.6)) responsible for the movement of lattice gas was equal to 0.0005 [lu]. This value of the external mass force was assigned to all nodes of the numerical grid belonging to a pore space. Relaxation time was constant and equal to 15. A different number of iterations was performed in every case. Calculations were terminated if the absolute difference between the average

values of the velocity modulus from two more iterations ($c_i = |v_{ave}^n - v_{ave}^{n-1}|$, where c_i is the convergence indicator and n is the number of iterations) was less than the significant digits for double precision literals. The calculated values of the convergence indicator for all simulations are shown in Fig. 5. Exemplary results of LBM calculations are presented in Fig. 6.

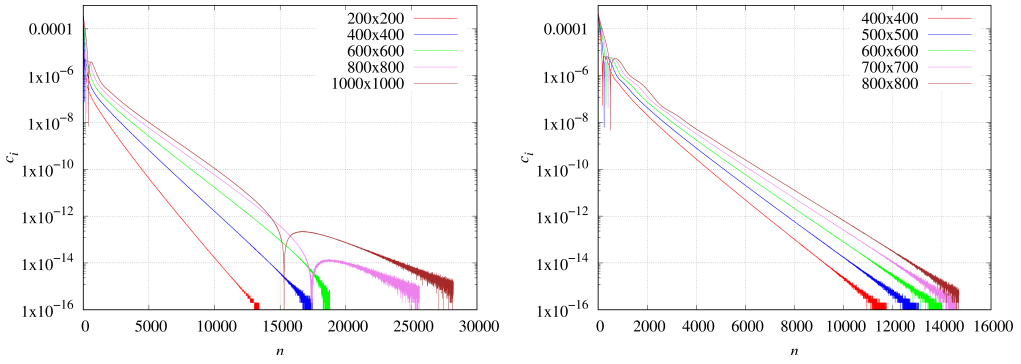

Fig. 5. History of the convergence indicator for all lattices and circle-based (left) and square-based (right) geometries.

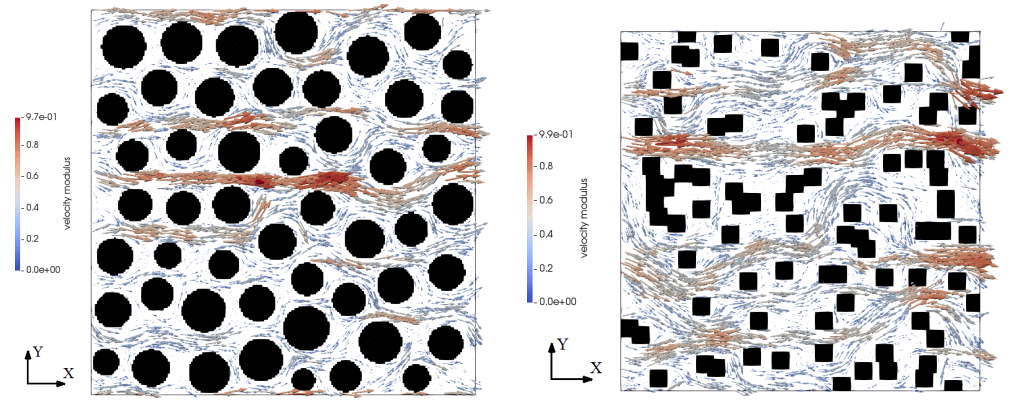

Fig. 6. Velocity field for a circle-based geometry with 400×400 grid resolution (left), and a square-based geometry with 400×400 grid resolution (right).

Taking into account the assumed time step and the spatial step, both equal to 1, the maximum velocity noted in any lattice node, in considered geometries was equal to 0.126 [m/s]. The kinematic viscosity for relaxation time used was about 4.8 [m^2/s]. By these values the Reynolds number did not exceed 0.005 in all calculations.

3.2. Determination of streamlines

The methodology presented in Subsection 2.4 may be applied when velocity fields are available. Another numerical code, where the discrete velocity fields obtained in LBM simulations were the input data, was developed and used in this stage. The streamlines calculated from every boundary node for the circle-based geometry with 400×400 lattice resolution are presented in Fig. 7. The results of the second simulation involving the same geometry, but only selected Starting Points, are presented in Fig. 8. The normalised velocity profile for the

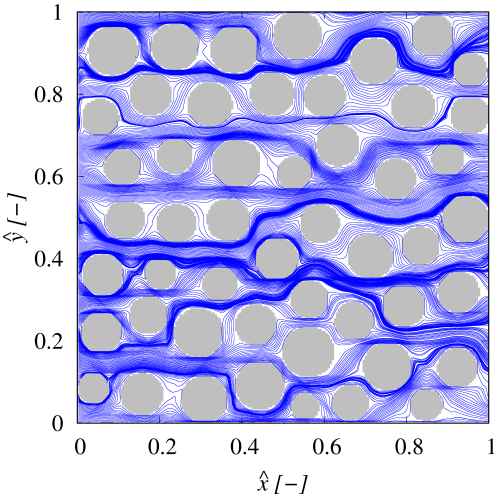

FIG. 7. All streamlines derived for a circle-based geometry with 400 × 400 grid resolution.

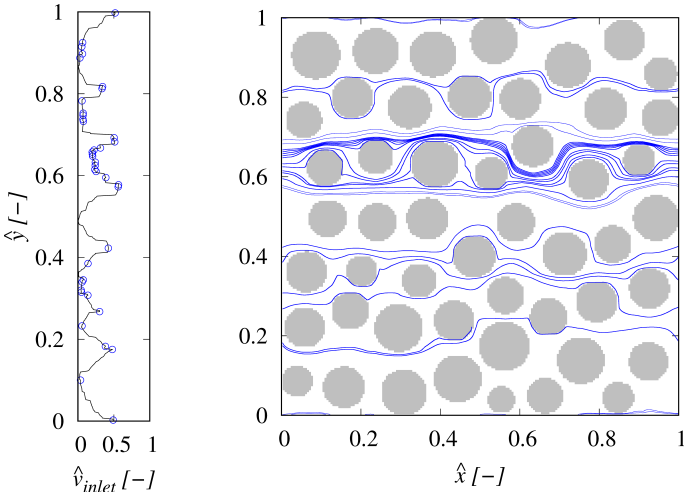

FIG. 8. Selected streamlines derived for a circle-based geometry with 400 × 400 grid resolution.

first column of nodes is shown on the left. All Starting Points (local maxima of the velocity profile) are marked with circles. A total of 25, 39, 40, 39 and 36 of Starting Points were used for circle-based geometries with 200×200, 400×400, 600×600, 800×800 and 1000×1000 resolution, respectively. The results for the square-based geometry are shown in Figs. 9 and 10. A total of 26, 24, 26, 22 and 20 Starting Points were used for squared-based geometries with 400×400,

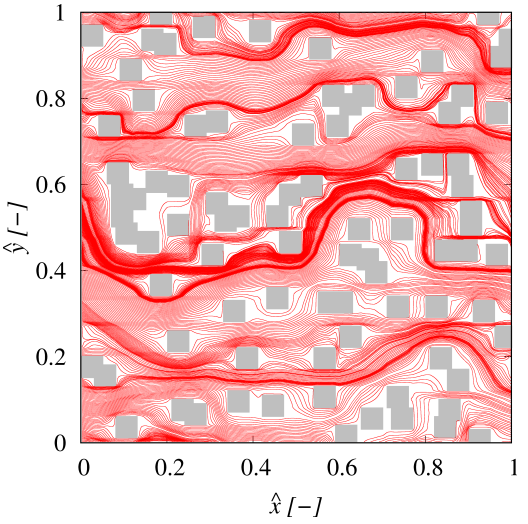

Fig. 9. All streamlines derived for a square-based geometry with 400×400 grid resolution.

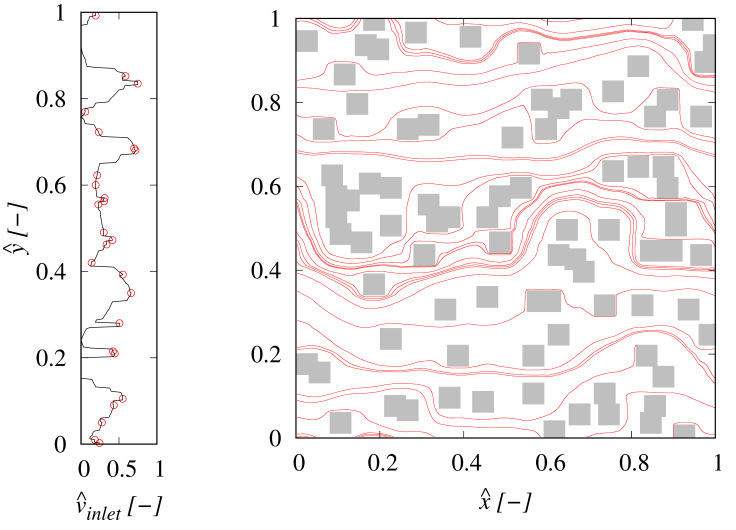

Fig. 10. Selected streamlines derived for a square-based geometry with 400×400 grid resolution.

500 × 500, 600 × 600, 700 × 700 and 800 × 800 resolution, respectively. The influence of the periodic boundary can be observed in all figures.

The data relating to the exceptions mentioned in Subsection 2.4 are presented in Table 2. Symbols n_{SP} and n_{CP} denote the number of Starting Points and the number of complete paths, respectively. The values of all Starting Points and selected Starting Points (separated by / symbol) are indicated in each cell. In some cells, the percentage of exceptions relating to the number of n_{SP} is given in parentheses. In some cases, the number of Starting Points was lower than the resolution in the Y direction because only Starting Points located in pore space were taken into account in path calculations. As expected, the number of exceptions was higher in the square-based geometry because the presence of overlapping obstacles contributed to the more complex shape of pore channels. Exceptions occurred mainly in the paths exiting from Starting Points other than SPs with local inlet velocity extrema.

Table 2. The number of exceptions in streamline calculations.

	Grid resolution	n_{SP}	n_{CP}	"reverse flow"	"dead end"	"looping"
Circle-based geometry: all SPs / selected SPs	200 × 200	200 / 25	125 / 18	39 (19.5%) / 3	36 (18.0%) / 4	0 / 0
	400 × 400	400 / 39	347 / 33	36 (9.0%) / 5	16 (4.0%) / 1	1 / 0
	600 × 600	600 / 40	544 / 33	40 (6.7%) / 5	16 (2.7%) / 2	0 / 0
	800 × 800	800 / 39	761 / 33	29 (3.6%) / 5	10 (1.3 %) / 1	0 / 0
	1000 × 1000	1000 / 36	955 / 32	35 (3.5%) / 4	10 (1.0%) / 0	0 / 0
Square-based geometry: all SPs / selected SPs	400 × 400	360 / 26	285 / 25	42 (11.1%) / 0	32 (8.4%) / 1	1 / 0
	500 × 500	450 / 24	377 / 22	49 (10.9%) / 1	23 (5.1%) / 1	1 / 0
	600 × 600	540 / 26	469 / 23	57 (10.6%) / 1	14 (2.6%) / 2	0 / 0
	700 × 700	630 / 22	544 / 20	64 (10.2%) / 1	21 (3.3%) / 1	1 / 0
	800 × 800	720 / 20	640 / 19	64 (8.9%) / 1	15 (2.1%) / 0	1 / 0

Several causes of the exceptions may be identified: a) discrete form of the input velocity field (dependent on the applied grid resolution); b) interpolation of the velocity field from lattice nodes to other locations; c) limited number of bits for data representation; d) rounding caused by data representation; e) calculations performed on very small numbers. Causes a) and e) play the key role.

Calculation problems always occur in locations where the velocity field tends to zero (when velocity components are relatively large, streamlines are calculated quickly and have clear directions). In addition to the rarest grid for a circle-based geometry, the percentage of "dead ends" is somewhat higher in square-based geometries. The problem of "reverse flows" has a different character and depends on the distribution of obstacles at the macro-scale. The percentage of "reverse flows" is higher because square-based geometries have a more complex

structure. The "looping" exception is relatively rarely encountered, but it is also more frequently observed in square-based geometries.

The streamlines derived with the use of selected Starting Points are compared in Fig. 11. The coordinates of all points forming individual streamlines were normalised to obtain one uniform domain with a size of 1×1. The streamlines calculated for different grid resolutions are similar, but not the same. The streamlines for the circle-based geometry with 400×400 grid resolution have a somewhat different shape than other streamlines. The normalised inlet veloc-

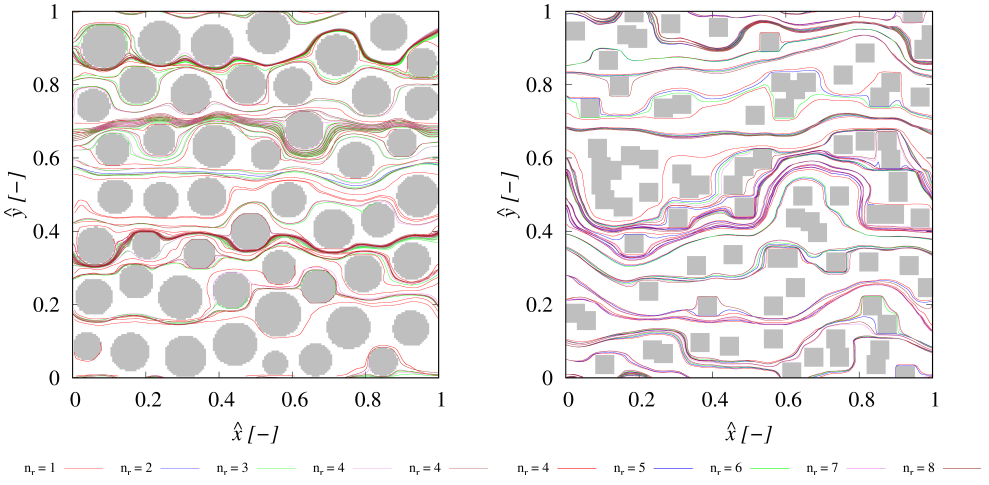

Fig. 11. Comparison of all streamlines for circle-based (left) and square-based (right) geometries.

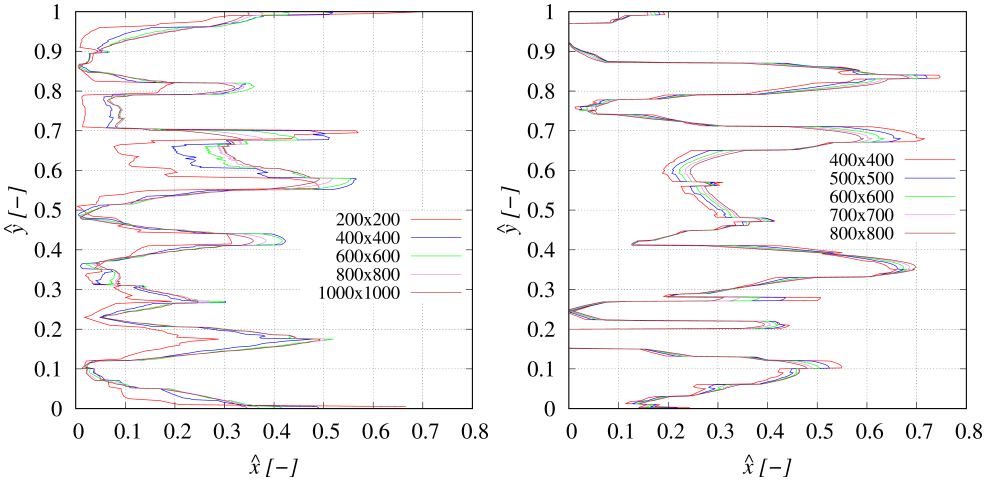

Fig. 12. Comparison of inlet velocity profiles for circle-based (left) and square-based (right) geometries.

ity profiles are compared in Fig. 12. In this case, the most outlying observation was also made in the circle-based geometry for the coarsest lattice. The data obtained for the square-based geometry were also very similar. The analysed cases did not differ significantly.

As expected, the method involving selected Starting Points required less computation time (Fig. 13). In all cases, pure calculation time appeared to increase linearly as a function of the number of lattice nodes. For higher resolutions, the increase in total computation time was probably caused by input-output operations needed to save larger files.

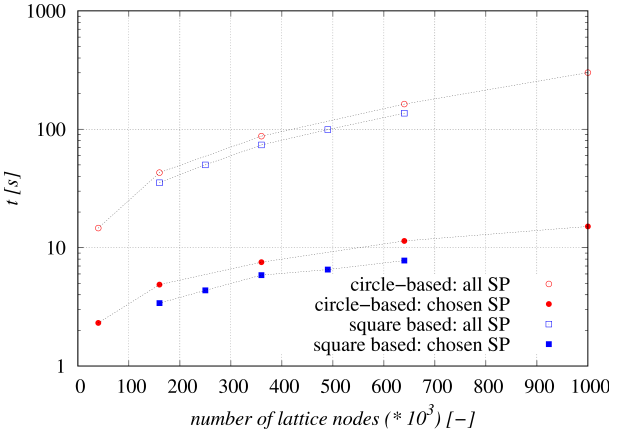

Fig. 13. Comparison of computation times.

The method involving selected Starting Points decreased computation time in circle-based geometries by 6.32, 8.82, 11.59, 14.34 and 19.92 for grid resolutions 200×200, 400×400, 600×600, 800×800 and 1000×1000, respectively. In turn, in the square-based geometry, computation time decreased by a factor of 10.4, 11.5, 13.0, 15.2 and 17.6 for grid resolutions 400×400, 500×500, 600×600, 700×700 and 800×800, respectively. The decrease in computation time is significant, and it constitutes the main advantage of the presented approach.

The available data for both geometries are compared in Figs. 14–15. Points represent the individual values of tortuosity calculated for all streamlines with known values of L_p. Grid resolutions are marked in different colours. In some cases, the individual values of tortuosity range far from the average values denoted by coloured solid lines. Streamline tortuosity often forms distinct groups with very similar values, which is a characteristic feature of the described approach. The above can be attributed to the fact that some neighbouring streamlines have very similar shapes, besides the section between the inlet line/plane and the first obstacle (see Fig. 11). A path that begins near the centre of an obstacle features more turns than a path which begins closer to the edge of the obstacle.

The results of streamline tortuosity (τ_s) calculations should be compared with the data from other approaches (Table 3). In the first step, hydraulic tortuosity (τ_h) was calculated for each available velocity field with the use of formula (1.2).

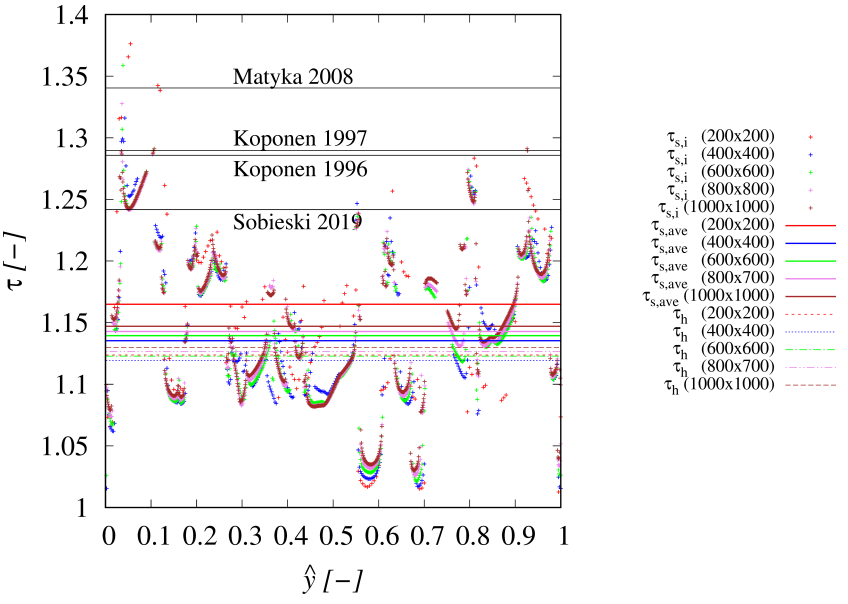

Fig. 14. Comparison of tortuosity for a circle-based geometry and all Starting Points.

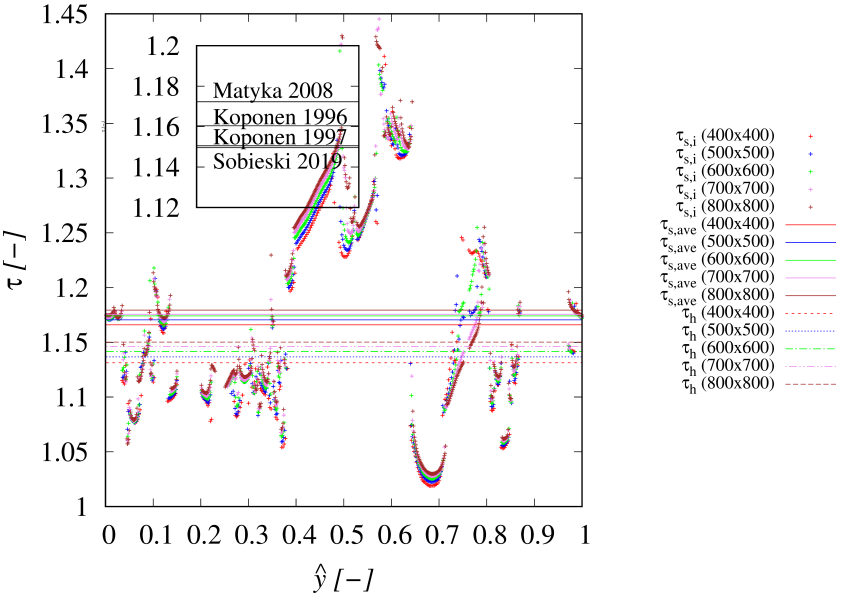

Fig. 15. Comparison of tortuosity for a square-based geometry and all Starting Points.

These values are represented by coloured dotted lines in Figs. 14 and 15. In general, hydraulic tortuosity was lower than streamline tortuosity. This trend is independent of a geometry type, and it is consistent with the results previously presented in [20]. The formulas shown in Table 1 (denoted by the same numbers in Tables 1 and 3) were applied in the second step. These formulas are independent of grid resolution or geometry type, and they always produce only one value. The values of tortuosity calculated with the use of empirical formulas are represented by black lines in Figs. 14 and 15. In Fig. 15, these lines coincide; therefore, they were plotted in a separate chart. The degree of convergence depends on the type of geometry. The presence of overlapping or non-overlapping obstacles appears to play the key role in this case. In square-based geometries, the values calculated with empirical formulas are highly similar to each other as well as to hydraulic tortuosity values. The reason for the above is that all formulas in Table 1 were developed based on LBM simulations (analyses of hydraulic tortuosity) for similar square-based geometries with overlapping obstacles. In circle-based geometries, empirical formulas always produce much higher values of tortuosity than the remaining methods. This divergence could be a general characteristic of pore structures with non-overlapping obstacles. In Table 3, relative errors are additionally presented in brackets in relation to hydraulic tortuosity. The relative errors for streamline tortuosity were calculated in relation to hydraulic tortuosity for the same lattice resolution. In empirical formulas, relative errors were calculated in relation to hydraulic tortuosity for the highest grid resolution.

3.3. Discussion

Ten lattices with different resolutions representing two porous structures were analysed in this study. The Lattice Boltzmann Method was applied to every lattice to simulate the creeping flow of lattice gas in pore channels. All simulations were performed under the same conditions and at the same level of convergence. After obtaining the velocity fields, streamlines were calculated in two variants. The shape of the normalised inlet velocity profiles was very similar in every case. The number of extrema and exceptions was also approximately constant, excluding in the circle-based geometry and the resolution 200×200. Despite the fact that all pore channels had an appropriate width, lattice resolution was still insufficient in this case.

Hydraulic and streamline tortuosity are compared in Fig. 16 as a function of the number of grid nodes. The values generated by both methods were shifted relative to each other, but they increased in a similar manner. The shift was caused by the characteristic features of the applied methodology. The application of all Starting Points in the calculations produces more consistent and smoothed

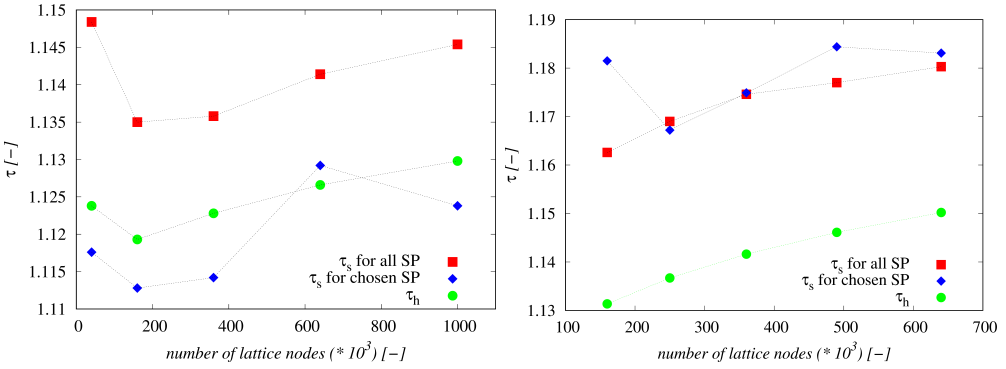

Fig. 16. Comparison of tortuosity values in the two methods applied to circle-based (left) and square-based (right) geometries.

results. However, streamline tortuosity for selected Starting Points may also be used if only one representative value is needed and a relative error approximating several percent is acceptable. The above approach was especially effective in the circle-based geometry, where a relative error did not exceed one percent for all lattice resolutions. It could be postulated that such an effect is characteristic of pore structures with non-overlapping obstacles, where long, complex and isolated channels do not occur. In fact, the exact value of tortuosity is unknown. In the presented calculations, hydraulic tortuosity was used as the reference value.

Table 3. Average values of streamline tortuosity, hydraulic tortuosity and tortuosity calculated with selected formulas.

	Circle-based geometry			Square-based geometry		
	grid res.	all SP	chosen SP	grid res.	all SP	chosen SP
τ_s	200 × 200	1.1484 (2.19%)	1.1176 (0.55%)	400 × 400	1.1626 (2.76%)	1.1815 (4.43%)
	400 × 400	1.1350 (1.40%)	1.1128 (0.58%)	500 × 500	1.1690 (2.84%)	1.1672 (2.68%)
	600 × 600	1.1358 (1.16%)	1.1142 (0.77%)	600 × 600	1.1746 (2.89%)	1.1749 (2.92%)
	800 × 800	1.1414 (1.31%)	1.1292 (0.23%)	700 × 700	1.1770 (2.70%)	1.1844 (3.34%)
	1000 × 1000	1.1454 (1.38%)	1.1238 (0.53%)	800 × 800	1.1803 (2.62%)	1.1831 (2.86%)
τ_h	200×200	**1.1238**		400 × 400	**1.1314**	
	400 × 400	**1.1193**		500 × 500	**1.1367**	
	600 × 600	**1.1228**		600 × 600	**1.1416**	
	800 × 800	**1.1266**		700 × 700	**1.1461**	
	1000 × 1000	**1.1298**		800 × 800	**1.1502**	
(1)	–	1.2858 (13.81%)		–	1.1604 (0.89%)	
(2)	–	1.2896 (14.14%)		–	1.1505 (0.03%)	
(3)	–	1.3404 (18.68%)		–	1.1723 (1.92%)	
(4)	–	1.2430 (10.02%)		–	1.1506 (0.03%)	

However, this is only an approximation, and the errors would be different if other types of tortuosity, such as diffusional, electrical or acoustic tortuosity, were to be adopted as the exact value. In fact, one or both of the analysed pore structures could be even better represented by streamline tortuosity.

It can also be concluded that the empirical formulas in Table 1 are not appropriate for porous structures other than those based on overlapping squares or rectangles. Some of these formulas, in particular Eq. (1) and (2), are regarded as universal equations in the literature. It seems that this issue should be emphasised in the literature.

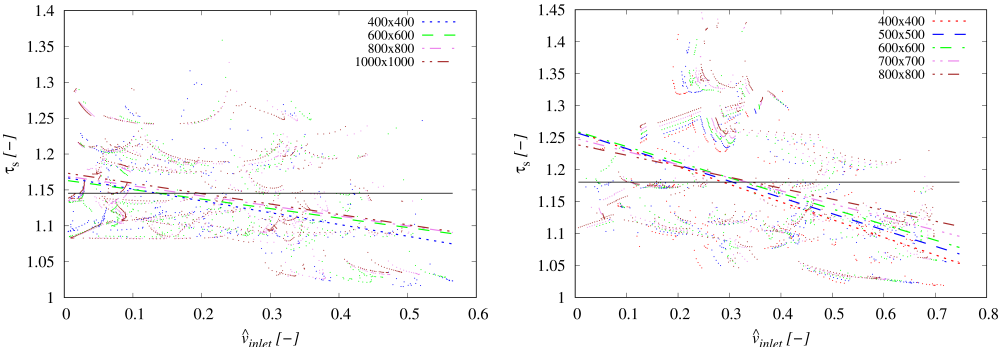

FIG. 17. The relationship between normalised inlet velocity and tortuosity for circle-based (left) and square-based (right) geometries.

The relationship between normalised inlet velocity and tortuosity for all Starting Points is shown in Fig. 17. The dotted line represents the linear trends of individual tortuosity values obtained for all grid resolutions. The fit was made using the least squares method. The black line represents the average tortuosity obtained for all Starting Points and the highest resolution. The shapes of inlet velocity profiles result directly from the geometry of pore channels. If the movement of lattice gas is not limited by obstacles, velocity increases and the streamlines are relatively straight. In such cases, tortuosity values are lower, and this trend is identical for both types of geometry. These trends are flattened when grid resolution increases. These observations suggest that in highly complex systems consisting of hundreds or thousands of obstacles, all paths will be so long that all local deviations will be averaged and streamline tortuosity will be independent of the local inlet velocity. Similar observations were made by other authors, including KOPONEN *et al.* [11]. The cited authors concluded that simulation quality depends on the relative scale between the size of the computational domain and obstacle size. In the context of the paper, it should be noted that the value of tortuosity may be somewhat underestimated in the method based on selected Starting Points, in particular in relatively small systems. However, underestimated values are closer to hydraulic tortuosity.

The distribution of streamline tortuosity for all analysed lattices is shown in Fig. 18. The average streamline tortuosity for the largest lattice is additionally represented by a black line. The presence of tortuosity ranges with a very small number of values is a characteristic feature of the proposed approach. This effect can probably be attributed to the small scale of the sample. Greater variation in the path length can be expected in larger porous systems. However, this could be a unique feature of specific porous media which is determined by, for example, the pore size distribution. This issue remains open. However, the relevant analyses should involve methods that generate many local tortuosity values. It should also be noted that the methodology based on hydraulic tortuosity and empirical formulas cannot be applied for this purpose.

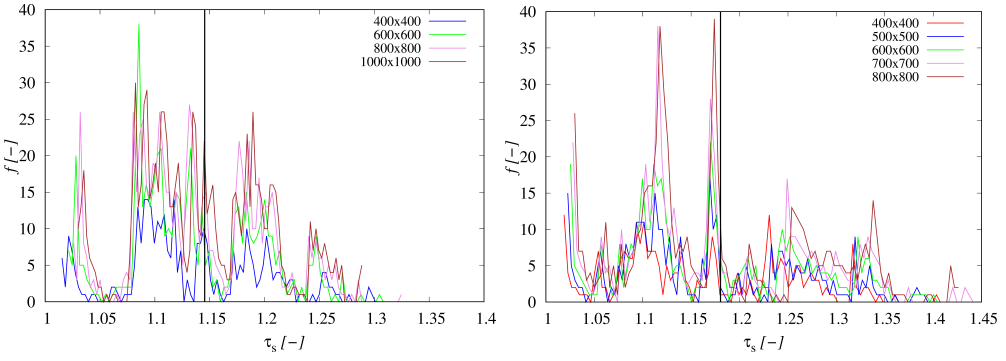

Fig. 18. Comparison of tortuosity distributions for circle-based (left) and square-based (right) geometries.

The performed study well illustrates the main difficulties associated with numerical analyses of porous structures. On the one hand, the analysed samples and grid resolution should be as large as possible. On the other hand, however, sample size and grid resolution cannot be too large due to limited computational power. In this context, the attempt to find methods that produce satisfactory results for larger systems while relying on the same computational power seems to be fully justified.

4. Summary

The following conclusions can be formulated based on the presented results:

- The calculation of streamlines exiting from selected Starting Points is a good alternative to calculations where all possible streamlines are taken into account. Computation time is significantly shorter and estimation quality is comparable. The proposed approach may be successfully applied in calculations based on other algorithms that generate numerous paths,

such as A-Star, Random Walking, Patch Searching Algorithm or the Path Tracking Method.

- The advantage of the streamline-based approach is that many individual values of tortuosity can be calculated, unlike in the methodology presented by Koponen et al., where only one value can be determined at a time. However, one tortuosity value representative of the entire porous medium is still available.
- The study demonstrated that the values of hydraulic as well as streamline tortuosity respond in similar manner to changes in network resolution. This observation confirms that grid resolution is an important consideration that is independent of the applied approach.
- The use of empirical formulas for tortuosity estimation is subject to great uncertainty and risk, that the obtained result is very different from the actual value. Particularly in cases, when any alternative data are not available.

Acknowledgements

This study was funded by the Polish Ministry of Science and Higher Education within the framework of the University's statutory research.

References

1. J. Kozeny, *On capillary conduction of water in soil*, Academy of Sciences in Vienna, Session Reports, **136/2a**, 271–306, 1927.
2. P.C. Carman, *Fluid flow through a granular bed*, Transactions of the Institute of Chemical Engineers, Jubilee Supplement, **75**, 32–48, 1997.
3. J. Bear, *Dynamics of Fluids in Porous Media*, Courier Dover Publications, New York, 1972.
4. C.F. Berg, *Permeability description by characteristic length. tortuosity. constriction and porosity*, Transport of Porous Media, **103**, 381–400, 2014.
5. J. Fu, H.R. Thomas, C. Li, *Tortuosity of porous media: image analysis and physical simulation*, Earth-Science Reviews, **212**, 103439, 1–30, 2021.
6. W. Sobieski, S. Lipiński, *The analysis of the relation between porosity and tortuosity in granular beds*, Technical Sciences, **20**, 75–85, 2017.
7. T.G. Zieliński, *Generation of random microstructures and prediction of sound velocity and absorption for open foams with spherical pores*, The Journal of the Acoustical Society of America, **137**, 1790–801, 2015.
8. A. Duda, Z. Koza, M. Matyka, *Hydraulic tortuosity in arbitrary porous media flow*, Physical Review E, **84**, 036319, 2011.

9. A. Nabovati, A.C.M. Sousa, *Fluid flow simulation in random porous media at pore level using Lattice Boltzmann Method*, in: F.G. Zhuang, J.C. Li [eds.], *New Trends in Fluid Mechanics Research*, Springer, Berlin, Heidelberg, 2007.

10. A. Koponen, M. Kataja, J. Timonen, *Permeability and effective porosity of porous media*, Physical Review E, **56**, 3319, 1997.

11. A. Koponen, M. Kataja, J. Timonen, *Tortuous flow in porous media*, Physical Review E, **54**, 406, 1996.

12. M. Matyka, Z. Koza, *How to calculate tortuosity easily?*, AIP Conference Proceedings, **17**, 1453, 2012.

13. W. Sobieski, *Numerical investigations of tortuosity in randomly generated pore structures*, Mathematics and Computers in Simulation, **166**, 1–20, 2019.

14. M. Matyka, A. Khalili, Z. Koza, *Tortuosity-porosity relation in porous media flow*, Physical Review E, **78**, 026306, 2008.

15. W. Sobieski, *Calculating tortuosity in a porous bed consisting of spherical particles with known sizes and distribution in space*, Research Report, **1**, 2009, Winnipeg, Canada, 2009.

16. W. Sobieski, Q. Zhang, C. Liu, *Predicting tortuosity for airflow through porous beds consisting of randomly packed spherical particles*, Transport in Porous Media, **93**, 431-445, 2012.

17. R.I. Al-Raousha, I.T. Madhounb, *TORT3D: A MATLAB code to compute geometric tortuosity from 3D images of unconsolidated porous media*, Powder Technology, **320**, 99–107, 2017.

18. K. Pearson, *The problem of the random walk*, Nature, **72**, 294, 1905.

19. P.E. Hart, N.J. Nilsson, B. Raphael, *A formal basis for the heuristic determination of minimum cost paths*, IEEE Transactions on Systems Science and Cybernetics, **4**, 100–107, 1968.

20. W. Sobieski, M. Matyka, J. Gołembiewski, S. Lipiński, *The Path Tracking Method as an alternative for tortuosity determination in granular beds*, Granular Matter, **20**, 72, 2018.

21. R. Eckhardt, *Stan Ulam, John von Neumann, and the Monte Carlo Method*, Los Alamos Science, **5**, 131–143, 1987.

22. M. Marek, *Numerical generation of a fixed bed structure*, Chemical and Process Engineering, **34**, 347–359, 2013.

23. M. Marek, *Numerical modeling of random packed beds of various packing densities with a sequential deposition algorithm*, AIP Conference Proceedings, 2078, 020015, 2019, doi.org/10.1063/1.5092018, published online: 04 March 2019.

24. U. Aaltosalmi, *Fluid Flow in Porous Media with the Lattice-Boltzman Method*, PhD Thesis, University of Jyväskylä, Finland, 2005.

25. M. Matyka, Z. Koza, J. Gołembiewski, M. Kostur, M. Januszewski, *Anisotropy of flow in stochastically generated porous media*, Physical Review E, **88**, 023018, 2013.

26. P.L. Bhatnagar, E.P. Gross, M. Krook, *A model for collisional processes in gases I: small amplitude processes in charged and neutral one component system*, Physical Review E, **94**, 511–524, 1954.

27. M.C. SUKOP, D.T. THORNE, *Lattice Boltzmann Modeling – An Introduction for Geoscientists and Engineers*, Springer, 2006.

28. P.-H. KAO, R.-J. YANG, *An investigation into curved and moving boundary treatments in the Lattice Boltzmann Method*, Journal of Computational Physics, **227**, 5671–5690, 2008.

29. A. KAJZER, J. POZORSKI, *Application of the Lattice Boltzmann Method to the flow past a sphere*, Journal of Theoretical and Applied Mechanics, **55**, 1091–1099, 2017.

Received August 11, 2021; revised version February 07, 2022.
Published online March 08, 2022.