Calculating tortuosity in a porous bed composed of spherical particles with known size and distribution in space

Research Report No. 1/2009

This document (release date: 14 January 2019) is a direct translation of the Polish version of the original research report developed in December 2009. The calculation method presented in this document has been termed the **Path Tracking Method** by the author, and the method is deployed by the **PathFinder** program. Detailed information about the method, the developed software and its applications is avaiable on the website of the PathFinder project at http://www.uwm.edu.pl/pathfinder/.

Wojciech Sobieski

University of Warmia and Mazury in Olsztyn, Faculty of Technical Sciences
University of Manitoba, Department of Biosystems Engineering

Abstract: The report describes the process of developing a computer program for calculating tortuosity in a porous medium consisting of spheres with a known diameter and position. It explains the key concepts, possible solutions to the problem and various issues related to the implementation of the developed method. Specific problems are described in successive chapters.

WINNIPEG (CANADA), December 2009

T A B L E O F C O N T E N T S

1. Introduction

The following values describe the properties of porous media:

- volumetric porosity e [m^3/m^3] which measures the fraction of the volume of voids in a porous medium [9]:

$$e = \lim_{\Delta V \to \Delta V_g} \frac{\Delta V_p}{\Delta V} \tag{1}$$

where V_p is total void volume [m^3], V is sample volume [m^3], and V_g is boundary volume [m^3]. Boundary volume has been introduced to ensure that the selected sample is sufficiently large and that porosity does not equal 0 or 1. Boundary volume has been selected correctly when its increase does not induce a significant increase in porosity. Effective porosity is also defined in some applications to account for the fact that free fluid flow is not possible in all void spaces.

- average particle diameter d_{ave} [m] (in beds composed of spherical or quasi-spherical particles). Average particle diameter represents a typical bed particle, and it is used in materials composed of particles with various diameters. For differently shaped particles, the average diameter is usually determined with the use of sphericity coefficient φ. Sphericity is always somewhat less than unity ($\varphi < 1$).

- tortuosity τ [m/m] – is the ratio of the actual path length L_p traveled by fluid through voids in a porous medium and section L_0 [m] of the measured path (Fig. 1),

$$\tau = \frac{L_p}{L_0}. \tag{2}$$

By definition, tortuosity is always higher than unity (but it can equal 1 in boundary cases). The space between spheres where L_p is calculated will be referred to as the "path".

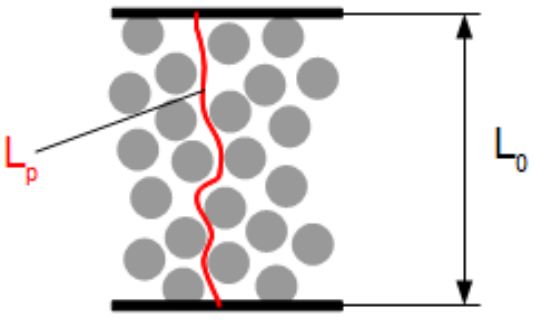

Figure 1. Visualization of tortuosity

2. Distribution of spherical particles in 3D space

In porous beds composed of spherical particles, particles are generally distributed in a non-uniform and a non-repeatable manner. This is because the position of every particle is determined by the local balance of forces: gravitational forces, force of reactions with other particles or container walls, and frictional forces. The only known information is that the first layer of particles will always come into direct contact with the bottom of the container. The fact whether all spheres have identical diameter is not relevant (although the problem is much more complex if spheres differ considerably in diameter).

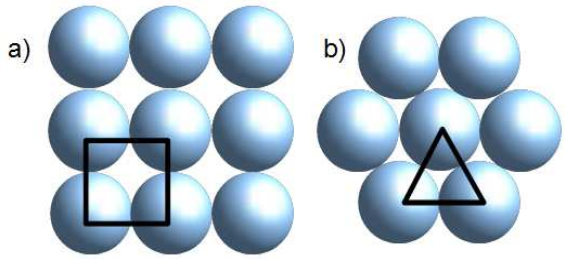

Figure 2. Theoretical distribution of spherical particles at the bottom of the container (top view).

In theory, particles that come into contact with the bottom of the container can be arranged in two principal ways. The distribution presented in Figure 2a can take place if the container has a rectangular cross-section and if the difference between particle diameter (which is always identical) and the length of the container wall is not significant. This difference should have a defined value to ensure that the particles are densely packed and are unable to move freely under the influence of reaction forces. This type of particle arrangement is generated artificially, and it generally does not occur in real systems composed of thousands or millions of particles. The above is validated by a simple experiment presented in Figure 3 which demonstrates that a triangle is the smallest geometric structure in a layer of spheres that are freely arranged on a plane. The geometric centers of the adjacent figures are the vortices of the triangle. The fact whether individual spheres come into contact and whether they have identical diameters is not relevant. The only important information is that individual spheres are always separated by void spaces.

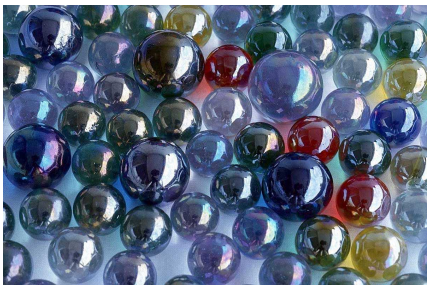

Figure 3. Actual distribution of spherical particles at the bottom of the container.

The arrangement of particles in successive layers poses another problem. Since the smallest geometric structure in a bed is a triangle (Fig. 4a), it can be assumed that the next sphere will be positioned directly above the triangle's center (Fig. 4c). If spheres have identical diameter and come into direct contact, their centers will form a tetrahedron in space. If the above conditions are not met, the spheres will form an irregular tetrahedron. Therefore, the tetrahedron is the basic arrangement pattern of particles with identical or similar diameter. A layered arrangement pattern (Fig. 4b) is improbable on a large scale, but it could be encountered locally in special cases.

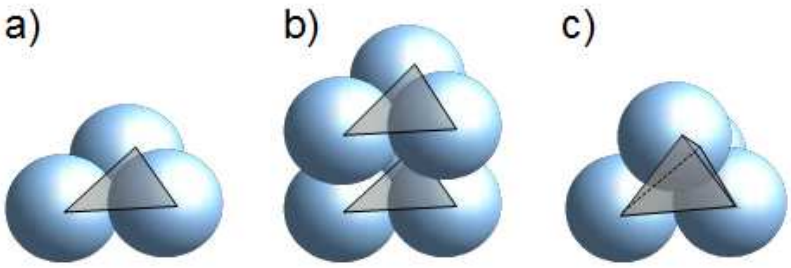

Figure 4. Distribution of spherical particles in a bed

3. Test bed

Tortuosity can be calculated with geometric methods only if detailed information about the position of a particle's center and the diameter of every particle is available. In this study, tortuosity was calculated based on the results of a simulation performed in the PFC3D program. This commercial program has been developed based on the Discrete Element Method. PFC3D supports analyses of particle motion in a porous bed under exposure to various forces, such as vibration. The program calculates the forces between adjacent particles and between particles and walls in view of the force of gravity and other body forces, subject to context. The coordinates of every sphere are calculated in every new time step. The program also monitors the relations between object to determine which spheres remain in contact in a given moment.

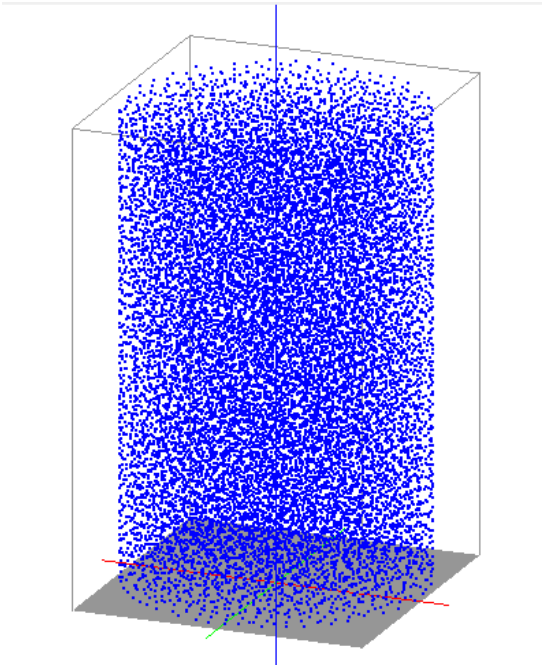

Figure 5. Visualization of the test bed

The tested porous bed is presented in Figure 5. Points represent the geometric centers of spheres. The bed was composed of 18188 spheres with an average diameter of 6.5 [mm] in a range of 5.5 to 7.5 [mm]. The container had a circular cross-section with diameter D = 0.15 [m]. The container had a height of 0.28 [m], but it was filled only to a height of 0.25 [m]. Detailed information about the discussed model and its validation was presented by [3,4]. The data generated by PFC3D included the numbers of all spheres and the coordinates of sphere centers.

4. Tortuosity calculation algorithm

The process of calculating tortuosity begins with the selection of a point, referred to as the "preliminary starting point", from which the path length will be determined. This point can be positioned arbitrarily, but it should not be positioned too close to the vertical wall of the container. In this study, the preliminary starting point was generally the point that was positioned centrally at the bottom of the container, an any exceptions to the above rule were implicitly stated. In Figure 6, the preliminary starting point was marked with coordinates (x_0, y_0, z_0).

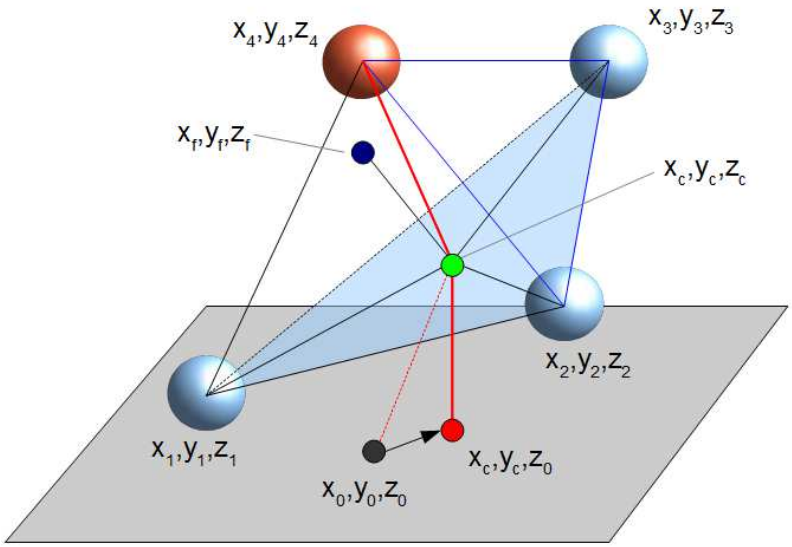

Figure 6. Method of calculating path length.

The preliminary starting point was calculated according to the below formula:

$$\begin{cases} x_0 = x_{\min} + \dfrac{x_{\max} - x_{\min}}{2} \\ y_0 = y_{\min} + \dfrac{y_{\max} - y_{\min}}{2}, \\ z_0 = z_{\min} - 0.5 \cdot d_{ave} \end{cases} \tag{3}$$

where: $x_{\min}$, $x_{\max}$, … - minimum and maximum values of the coordinate in each direction [m], d_{ave} - average diameter of all spheres in the bed [m]. In the next step, the numbers of the spheres forming the nearest triangle were determined. Sphere numbers ranged from 1 to 18188. The above numbers were used to determine vortex coordinates (numbered 1 to 3 in every run of the calculation loop) and the coordinates denoting the center of the triangle:

$$\begin{cases} x_c = \dfrac{x_1 + x_2 + x_3}{3} \\ y_c = \dfrac{y_1 + y_2 + y_3}{3}. \\ z_c = \dfrac{z_1 + z_2 + z_3}{3} \end{cases} \tag{4}$$

The triangle center with coordinates (x_c, y_c, z_c) is the second point of the path. The first path point has coordinates (x_c, y_c, z_0). This modification was applied to position the first section of the path perpendicularly to the bottom of the container and to prevent an unnecessary increase in path length. The last section of the path was modified accordingly. Point (x_c, y_c, z_0) was referred to as the "final starting point".

The method of calculating z_0 coordinate should be discussed in greater detail. As mentioned earlier, the data generated by PFC3D contained only sphere numbers and the coordinates of sphere centers. Initially, all spheres were sorted by height to determine the coordinate at the bottom of the container and the arbitrary coordinate on the top layer of the bed. It should be noted that the relations between sphere coordinates and sphere numbers generated by PFC3D were not modified. After this operation, the sphere with the smallest value of Z-coordinate ($z_{\min} = z_1$) was the first sphere to be saved in computer memory. By the same token, the topmost sphere ($z_{\max} = z_{n_s}$) was saved last. The value of z_0 coordinate was determined by subtracting the average sphere diameter (refer to formula 3) from the Z-coordinate of the first sphere. The same data were used to determine the length of path L_0 [m] where tortuosity is calculated:

$$L_0 = z_{n_s} - z_1 + d_{ave}\,, \tag{5}$$

where: z_{n_s} - Z-coordinate of the topmost sphere in the bed [m], z_1 - Z-coordinate of the bottommost sphere in the bed [m], n_s - number of spheres in the bed [-].

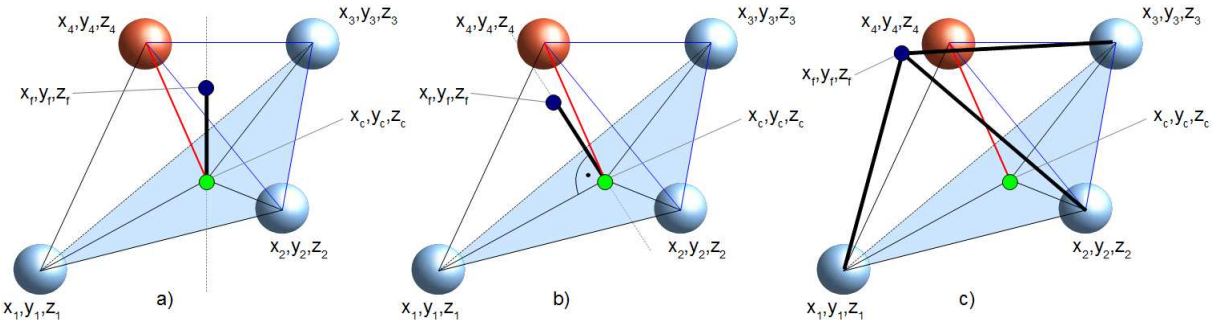

Figure 7. Methods of calculating the search point

The statement that a tetrahedron is the basic pattern of sphere arrangement was used in subsequent steps of the calculation process. Based on a triangle with vortex coordinates (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3), attempts were made to find a fourth sphere, where the resulting figure would most closely resemble a tetrahedron. The search for the fourth vortex began by estimating the position of the next sphere, and this position was referred to as the "search point" with coordinates (x_f, y_f, z_f). The coordinates of the actual sphere(x_4, y_4, z_4) positioned most closely to the search point were then determined. The method of determining search point coordinates proved to be highly significant for the discussed problem. Three such methods were developed:

The first method (Fig. 7a), referred to as the triangle center method (TCM), involves the calculation of the coordinates of the point positioned exactly above the center of the current triangle at a distance of:

$$h = h_0 \cdot C_h \cdot \sqrt{2/3}\,, \tag{6}$$

where: h - height of the tetrahedron [m], h_0 - specific height [m], C_h - correction factor [-] ($0 < C_h < 1$). Specific height h_0 can be the average side length of the current triangle l_{ave} [m] or the average sphere diameter d_{ave} [m]. The average diameter of spheres forming a given triangle can be potentially used. If $C_h = 1$, then h corresponds to the height of a tetrahedron with side length h_0.

Therefore, the search point is calculated with the use of the following formula:

$$\begin{cases} x_f = x_c \\ y_f = y_c \\ z_f = z_c + h_0 \cdot C_h \cdot \sqrt{2/3} \end{cases} . \tag{7}$$

In the first method, the path reaches the top layer of the bed in a rapid and "vertical" manner. This problem will be discussed in successive parts of the article. In this approach, path length is relatively small.

In the second method (Fig. 7b), referred to as the triangle normal method (TNM), the search point is positioned on the triangle normal at distance h. The cross product of vectors $\vec{w}$ and $\vec{u}$, obtained by subtracting the corresponding coordinates of two triangle points, was applied:

$$\begin{cases} w_x = x_2 - x_1 \\ w_y = y_2 - y_1 \\ w_z = z_2 - z_1 \end{cases}, \quad \begin{cases} u_x = x_3 - x_1 \\ u_y = y_3 - y_1 \\ u_z = z_3 - z_1 \end{cases}. \tag{8}$$

The direction cosines of the normal vector were calculated from the following dependencies:

$$\begin{cases} \alpha = w_y \cdot u_z - u_y \cdot w_z \\ \beta = u_x \cdot w_z - w_x \cdot u_z \\ \gamma = w_x \cdot u_y - u_x \cdot w_y \end{cases}. \tag{9}$$

The resulting vector was normalized:

$$\begin{cases} \alpha = \alpha / \delta \\ \beta = \beta / \delta \\ \gamma = \gamma / \delta \end{cases}, \tag{10}$$

where

$$\delta = \sqrt{\alpha^2 + \beta^2 + \gamma^2} . \tag{11}$$

The final formula for calculating the coordinates of the search point is:

$$\begin{cases} x_f = x_c + \alpha \cdot h_0 \cdot C_h \cdot \sqrt{2/3} \\ y_f = y_c + \beta \cdot h_0 \cdot C_h \cdot \sqrt{2/3} \\ z_f = z_c + \gamma \cdot C_\gamma \cdot h_0 \cdot C_h \cdot \sqrt{2/3} \end{cases}, \tag{12}$$

where C_γ is a dimensionless coordinate which shifts the triangle normal in the positive direction of the Z-axis. This default value of C_γ is 1. The coordinate has been applied to perform additional tests which will be described in successive parts of the article. Similarly to the first method, C_h was used to correct the distance between the point and the plane of the triangle.

The direction of the triangle normal is determined by the sequence in which the points forming vectors $\vec{w}$ and $\vec{u}$ [1] are defined. The aim of the simulation was to find a path from the bottom of the container to the top layer of the bed; therefore, triangle normals with a direction opposite to the Z-axis cannot be taken into consideration (the path would return towards the bottom of the container). For this reason, we have to determine which of the two points is positioned higher: the center of the triangle or the search point. If the search point is positioned below the center of the current triangle, we have to return to formula (8) and reverse point numbers in one of the vectors. The remaining calculations should then be repeated.

The third method (Fig. 7c), referred to as the ideal tetrahedron method (ITM), postulates the existence of a local tetrahedron where all sides are equal and have length h_0. The coordinates of the tetrahedron's vortices are described by the following formulas:

$$\begin{cases} h_0 \cdot C_h = \sqrt{(x_f - x_1)^2 + (y_f - y_1)^2 + (z_f - z_1)^2} \\ h_0 \cdot C_h = \sqrt{(x_f - x_2)^2 + (y_f - y_2)^2 + (z_f - z_2)^2} \\ h_0 \cdot C_h = \sqrt{(x_f - x_3)^2 + (y_f - y_3)^2 + (z_f - z_3)^2} \end{cases}. \tag{13}$$

The above system of nonlinear equations was solved with two numerical methods. The first method relies on the secant method, whereas the second method is based on a known value of the Z-coordinate, which considerably simplifies the system. However, neither of the algorithms was successfully implemented, and the search point was calculated erroneously in both cases. The triangle normal method proved to be a highly satisfactory approach; therefore, the above algorithms were not corrected, and ITM (and TCM) was ultimately rejected.

The number of the closest sphere is determined based on the coordinates of the search point. This part of the algorithm had to be defined for every special flag cell, i.e. a logical variable indicating whether a given sphere had been previously used to generate a triangle. The relevant information is required to prevent the program from classifying one of the spheres forming the current triangle as the closest sphere. If the above information were not available, the 3D figure would not be generated and the algorithm would not be continued. Every time a sphere is used in the algorithm, its flag is modified, and that sphere cannot be used again.

[1] I would like to thank Anna Trykozko of the University of Warsaw for her valuable input to solving the problem.

The calculated coordinates of tetrahedron vertices form three new triangles (1, 2, 3), (2, 3, 4) and (3, 1, 4). The triangle to be intersected by the path has to be selected. Initially, attempts were made to develop a method for selecting a triangle based on criteria such as the largest area, triangle center with the topmost position, triangle center positioned closest to the vertical line from the final starting point, and other criteria. However, numerous implementations and tests revealed that one of the triangle's vertices is always copied beginning from the first iteration. In other words, every successive triangle contained the same sphere despite the fact that the remaining spheres were changed. The above led to the rapid degradation of the algorithm and an infinite loop that did not produce a sensible solution. As a result, the calculation process was often disrupted. An example of the above is presented in Figure 8 where the selection criterion was the largest triangle area. The path was very long and convoluted (left) and triangle area increased gradually (right).

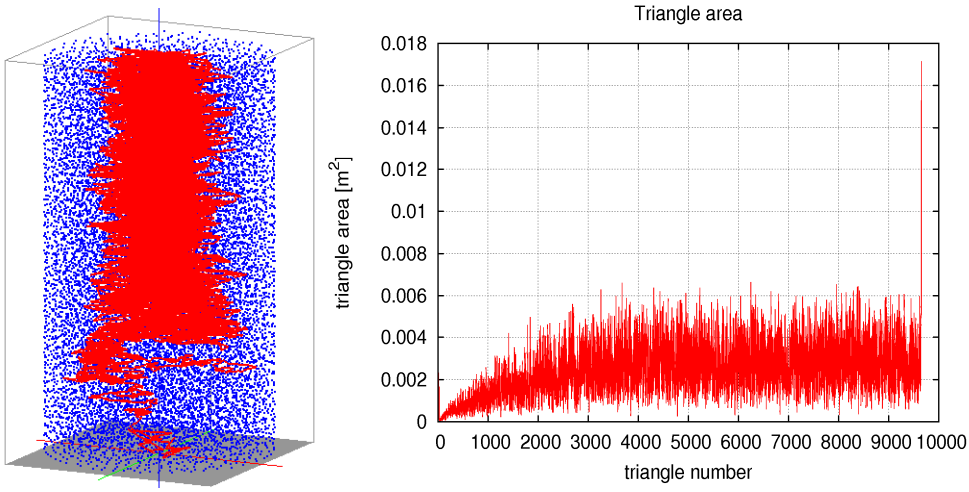

Figure 8. Examples of algorithm degradation

Algorithm degradation can be explained based on Figure 9. In this case, triangle II(124) has the largest area in the first iteration because one of its sides stretches between the bottommost sphere (1) and the topmost sphere (4). The next sphere will be positioned above the remaining spheres (due to TCM, TNM and ITM rules); therefore, triangle III(145) generated in the next iteration will have an even longer side (15). The resulting triangle will also have the largest area, and the next triangle will also use sphere number 1.

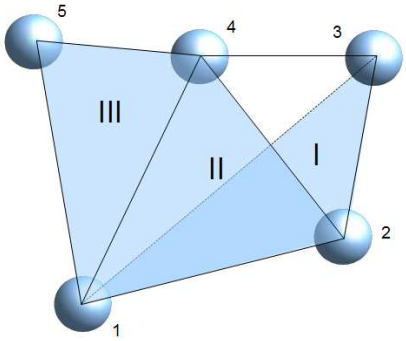

Figure 9. Visualization of the mechanism behind algorithm degradation

In view of the above, the bottommost sphere was ultimately rejected in every iteration. The three remaining spheres were used to generate a new triangle, and the search process was repeated until the achievement of a point where the distance between the search point and the top layer of the bed was smaller than the distance between that point and the nearest "free" sphere. If the above

condition was met, the loop was terminated, and the last point on the top layer of the bed, positioned directly above the center of the last triangle, was added to the path. The results of the calculations for the first two methods are presented in Figure 10.

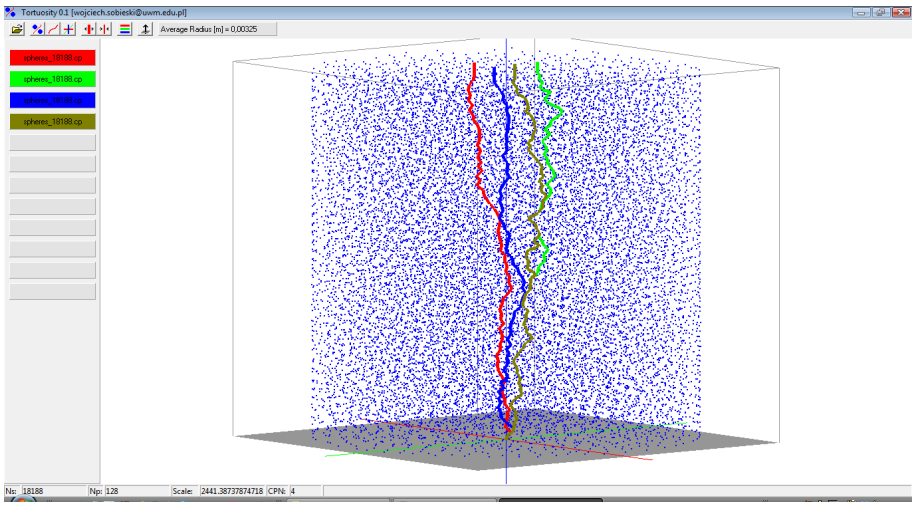

Figure 10. Visualization of results:
red: TCM, $h_0 = l_{ave}$; green: TNM, $h_0 = l_{ave}$;
blue: TCM, $h_0 = d_{ave}$; brown: TNM, $h_0 = d_{ave}$;

5. Implementation

The discussed algorithm was used to develop a computer program. The computation module was written in Fortran 90, and the code was compiled with gfortran. The computational area in every iteration was visualized with the Gnuplot interpreter based on the scripts generated automatically by the program. An example of data visualization is presented in Figure 11 which shows the results of the calculations performed in the first iteration for TNM, $h_0 = l_{ave}$ and correction factors equal to 1. The visualization includes the local fragment of space with the current triangle (green), the triangle center (red point which is also the second point on the path), the search point (filled green point) and the tetrahedral fourth comer (large blue point). The visualization presents the beginning of the calculation process, and starting points are also shown. The plot generated by Gnuplot can be rotated and enlarged, and the calculation process can be continued by pressing a key in a separate message window. The three successive steps of the same computational cycle are presented in Figures 12-14. The actual spheres are positioned very closely to the search point.

It should also be noted that the plots generated by Gnuplot are not set to scale, which is why they appear skewed, and the actual distances and angles can be significantly distorted. The above posed a challenge, and an alternative approach had to be used to test the quality of calculations and visualizations. Tetrahedron plots generated by Gnuplot and Gambit are presented in Figure 15[2].

The visualization mode (plots are/are not displayed for every iteration) is selected indirectly in the program. Regardless of the selected mode, the final path diagram with the surrounding spheres is displayed at the end of the calculation process. The final path for the discussed example is presented in Figure 16. The plot can be rotated and/or enlarged to analyze the path geometry in greater detail, as shown in Figure 17.

[2] Data were visualized in Gambit by Anna Trykozko of the University of Warsaw.

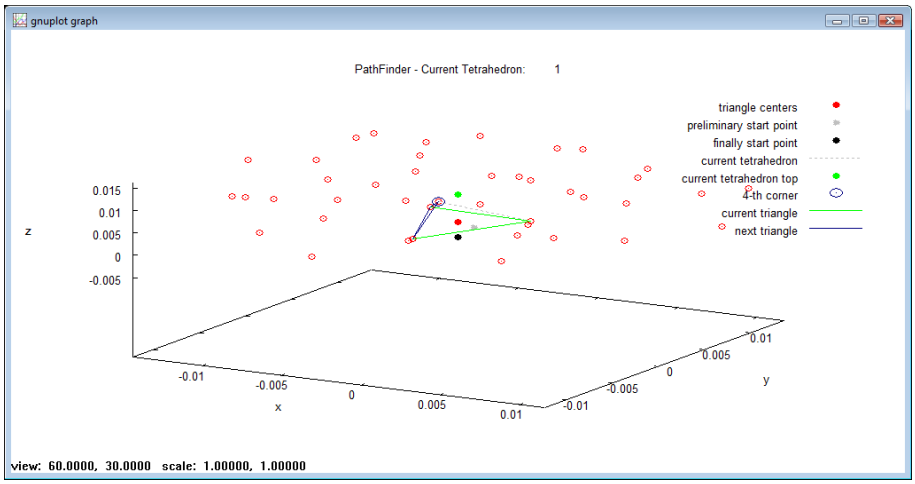

Figure 11. Visualization of the calculations in TNM for $h_0 = l_{ave}$ – iteration 1

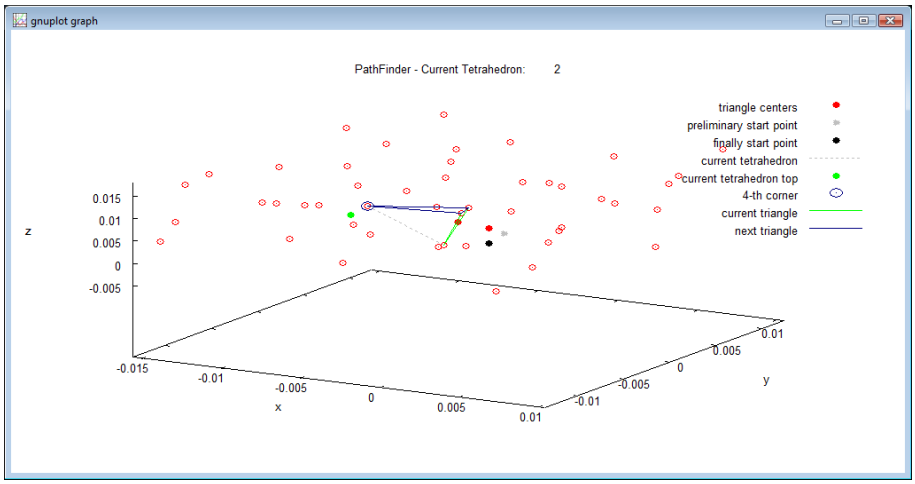

Figure 12. Visualization of the calculations in TNM for $h_0 = l_{ave}$ – iteration 2

Important information is displayed in the program window (Fig. 18). At the beginning, the program displays the name of the task and the number of spheres in the bed. During iterations, the number of the iteration and the numbers of the spheres forming the processed tetrahedron are displayed. At the end of the calculation process, the program generates the final report with information about the number of path points, bed height (L_0 [m]), path length (L_p [m]), tortuosity (τ [-]), bulk volume, internal area of the solid body, specific area of the porous body, volume of the porous body and porosity (e [-]). The configuration of computational methods and the values of the applied correction factors are also displayed. The last part of the final report presents the values of the terms of the Ergun equation (E_A – linear term, E_B – nonlinear term) and the Kozeny-Carman equation (KC). The value of the constant was not taken into account in the Kozeny-Carman equation.

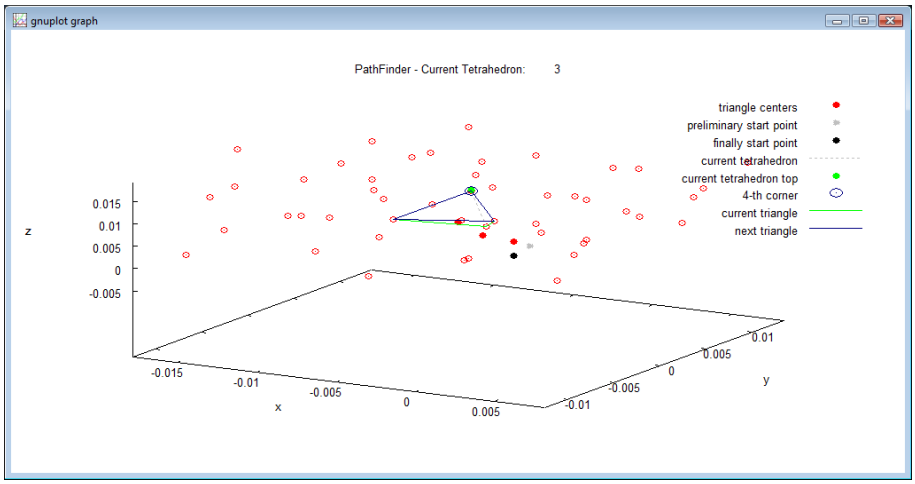

Figure 13. Visualization of the calculations in TNM for $h_0 = l_{ave}$ – iteration 3

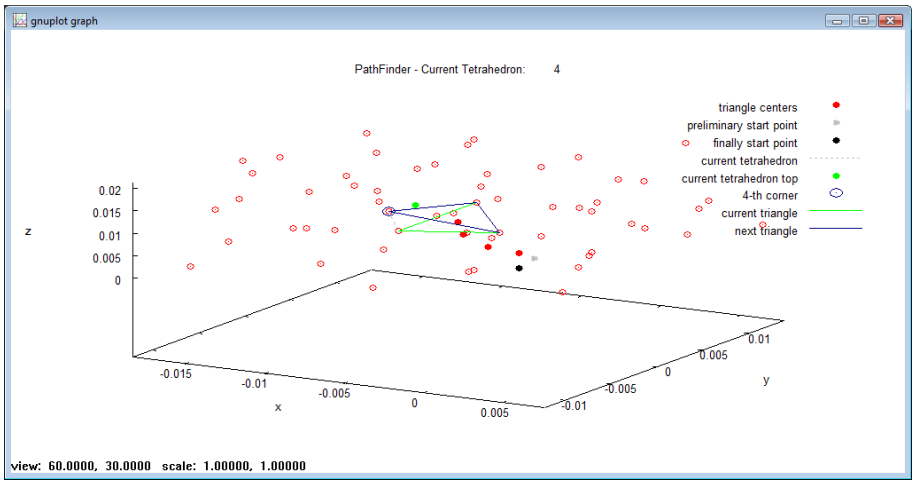

Figure 14. Visualization of the calculations in TNM for $h_0 = l_{ave}$ – iteration 4

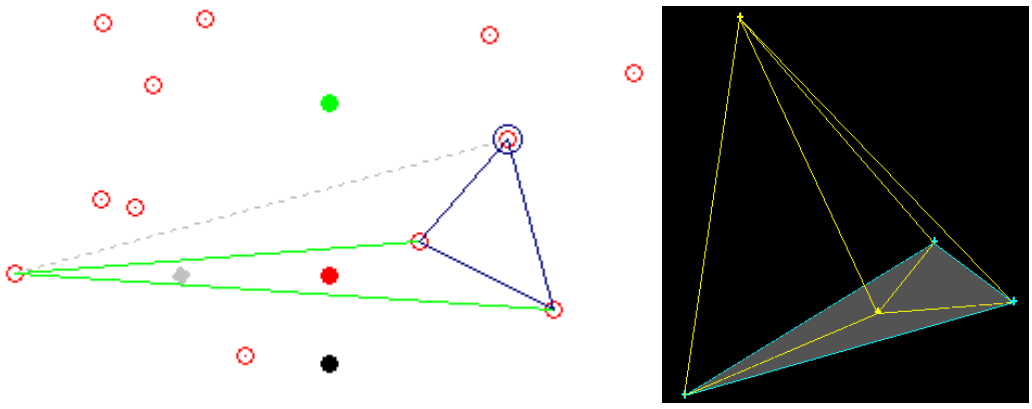

Figure 15. Visualizations of the same tetrahedron:
Gnuplot (left) and Gambit (right)

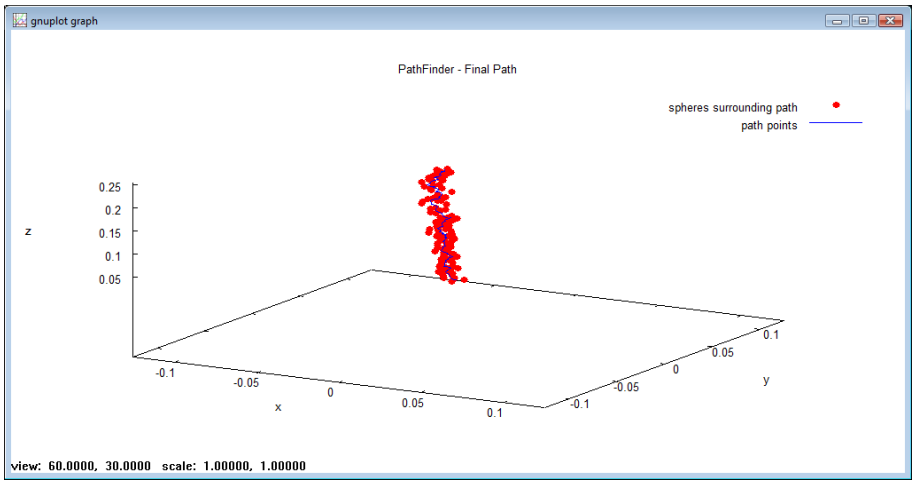

Figure 16. Visualization of the path calculated in TNM at $h_0 = l_{ave}$

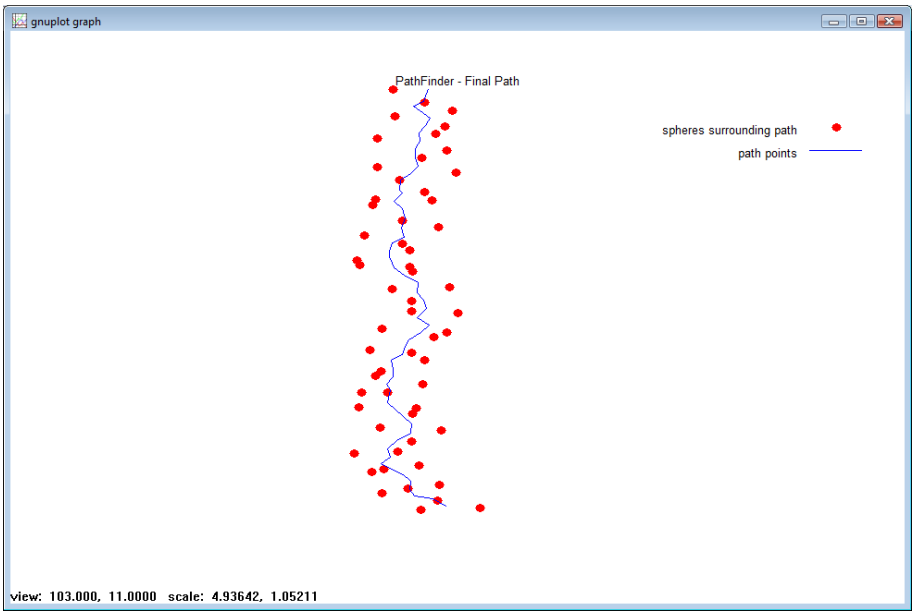

Figure 17. Visualization of the path calculated in TNM at $h_0 = l_{ave}$ – enlargement

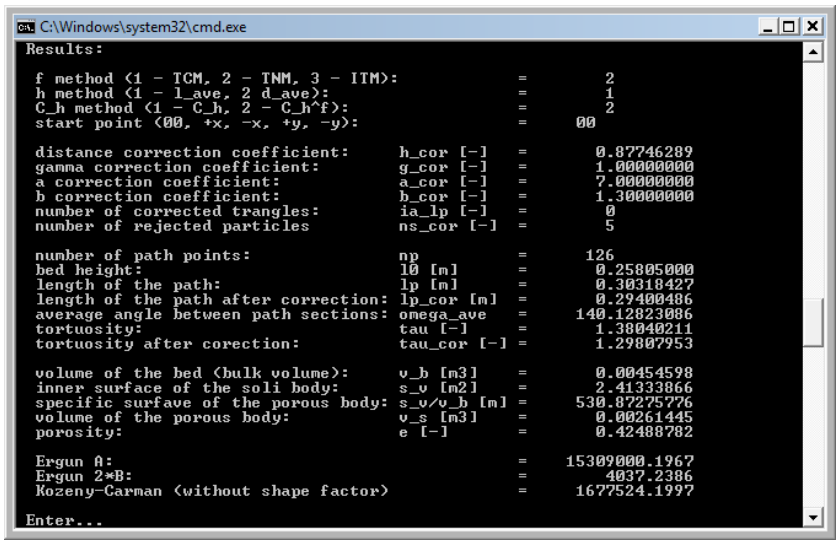

Figure 18. Program window – final report in TNM for $h_0 = l_{ave}$

The visualization of all spheres in the test bed was relatively slow in Gnuplot; therefore, an additional graphic postprocessor was developed in Borland Delphi 7 with the OpenGL graphic library. The bed and the paths (up to 12) are rapidly displayed in the program. The program window is presented in Figure 18.

6. Indicators of triangle geometry

The quality of the calculated path had to be analyzed in the process of developing the program. Triangle area and perimeter proved to be highly useful indicators. The indicator of triangle area I_A was defined as the ratio of the area of the current triangle A_i [m^2] to the area of an equilateral triangle A_0 [m^2] formed by three adjacent spheres with diameter d_{ave}:

$$I_A = \frac{A_i}{A_0}, \tag{14}$$

where

$$A_0 = d_{ave}^2 \cdot \frac{\sqrt{3}}{4}, \tag{15}$$

and

$$A_i = \sqrt{\frac{L}{2} \cdot \left(\frac{L}{2} - a\right) \cdot \left(\frac{L}{2} - b\right) \cdot \left(\frac{L}{2} - c\right)}. \tag{16}$$

Formula (16) is Heron's formula where L is the perimeter of a triangle, and a, b and c denote the lengths of triangle sides. All of the analyzed parameters were determined with the use of the formula for calculating the distance between two points in space:

$$l_{ij} = \sqrt{\left(x_i - x_j\right) + \left(y_i - y_j\right) + \left(z_i - z_j\right)}. \tag{17}$$

This approach was also used to calculate the perimeter indicator I_L:

$$I_L = \frac{L_i}{L_0}, \tag{18}$$

where: L_i - perimeter of the current triangle [m], perimeter of an equilateral triangle formed by three adjacent spheres with diameter d_{ave}.

The last parameter, which can be helpful for analyzing solutions in the TNM model, was the angle between the normal of the current triangle and the Z-axis (Fig. 19). The angle was calculated with the below formula:

$$I_\gamma = \arccos(\gamma), \tag{19}$$

and it was converted to degrees.

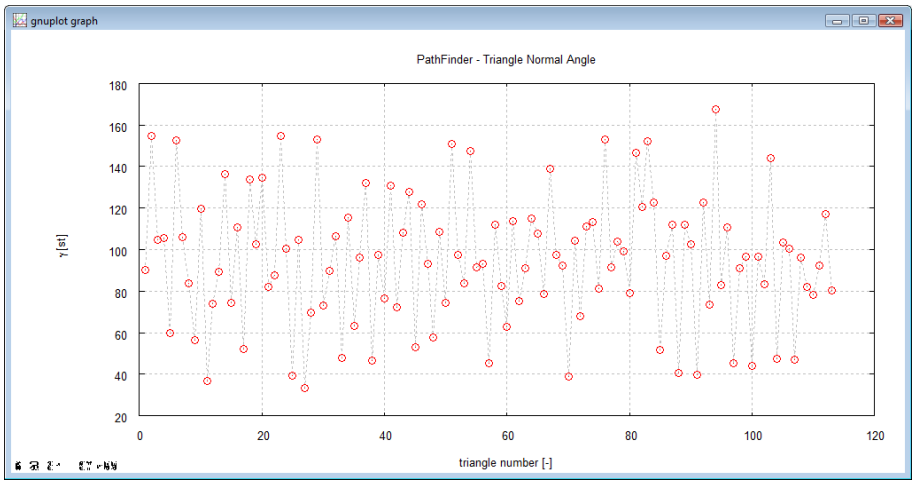

Figure 19. Exemplary distribution of I_γ in TNM for $h_0 = l_{ave}$

During the calculations performed for TNM and $h_0 = l_{ave}$, the algorithm was disrupted by an anomaly which occurred at roughly $2/3$ of bed height. This problem was discussed in greater detail in section 13. Beginning from this point, the area of every successive triangle increased (Fig. 20). Such a significant increase (up to several dozen times) in triangle area was not acceptable; therefore, an additional procedure was introduced to correct the points of the current triangle when its area exceeded the critical value A_{cr}. The critical value was achieved when the area of an ideal (equilateral) triangle was enlarged to an extent which enabled the next sphere with diameter d_{ave} to be positioned in its center. This problem is presented in Figure 21.

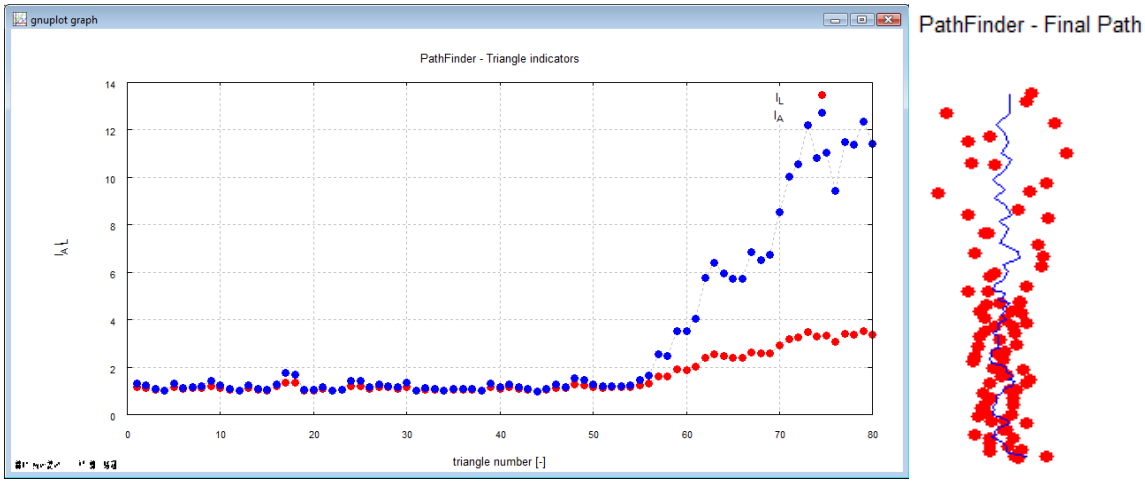

Figure 20. Disruption of the computational algorithm in TNM for $h_0 = l_{ave}$

In the system presented in Figure 21:

$$A_0 = \frac{\sqrt{3}}{4} \cdot d_{ave}^2 \,, \tag{20}$$

and

$$A_{cr} = 3 \cdot \frac{\sqrt{3}}{4} \cdot d_{ave}^2 , \tag{21}$$

therefore, the critical indicator of triangle area :

$$I_A^{cr} = \frac{3 \cdot \frac{\sqrt{3}}{4} \cdot d_{ave}^2}{\frac{\sqrt{3}}{4} \cdot d_{ave}^2} = 3 . \tag{22}$$

If the indicator of triangle area is equal or greater than I_A^{cr} in any step, the three spheres that are closest to the center of the existing triangle are identified, and sphere numbers in the current triangle are modified. The algorithm is repeated (to calculate the coordinates of vortices, the center, etc.) and then continued. The total number of corrections is presented in the final report.

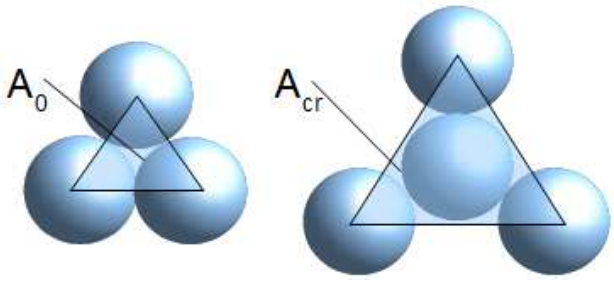

Figure 21. Illustration for the critical triangle problem.

The correction significantly improved path quality in TNM (Fig. 22), despite the fact that only 5 triangles were corrected. A local increase in triangle area was still observed (where $I_A < 3$ in every case), which could probably be attributed to less densely packed particles in this region. The above findings are generally consistent with the PVF3D simulation where minor local changes in porosity were observed throughout the entire bed (refer to section 13).

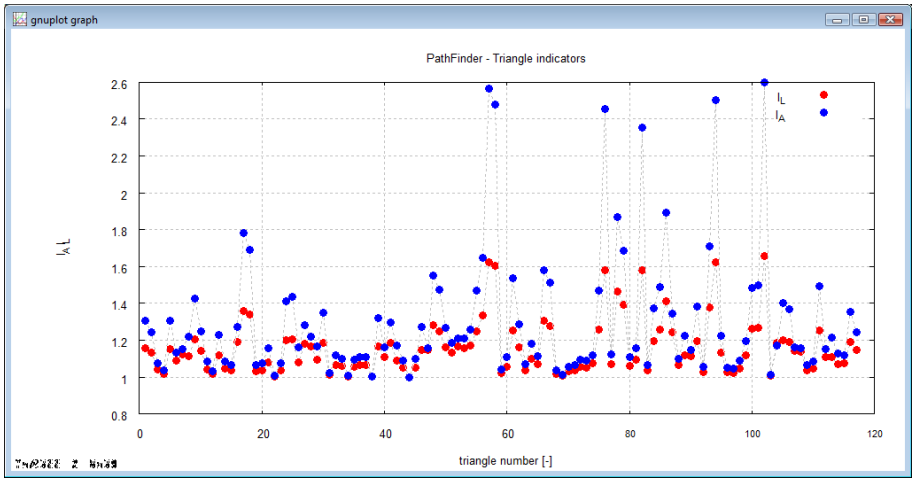

Figure 22. Distribution of indicator I_A in TNM for $h_0 = l_{ave}$ after correction

The results of the calculations for the four described methods are presented in Table 1. Paths are visualized in Figure 10. The results of the calculations for TNM and $h_0 = l_{ave}$ account for the corrections based on I_A^{cr}.

Table 1. Calculation results

Method	h_0	C_h	C_γ	n_p	L_0	L_p	τ	E_A	E_B	KC
TCM	l_{ave}	1.0	-	74	0.25805	0.28406	1.21177	15049234.3	3980	1439430
TCM	d_{ave}	1.0	-	83	0.25805	0.28715	1.23826	15049234.3	3980	1470903
TNM	l_{ave}	1.0	1.0	119	0.25805	0.31867	1.52501	15049234.3	3980	1811536
TNM	d_{ave}	1.0	1.0	128	0.25805	0.31531	1.49299	15049234.3	3980	1773498

7. Indicator of path quality

Indicator of path quality I_F was introduced to determine the effectiveness of the method of calculating the search point. This indicator evaluates the proximity of the current search point to the center of an actual sphere. Indicator I_F is expressed as follows:

$$I_F = \frac{\sqrt{(x_f - x_4)^2 + (y_f - y_4)^2 + (z_f - z_4)^2}}{0.5 \cdot d_{ave}}. \tag{23}$$

If I_F is less than 1, the search point is positioned inside an actual sphere (the distance to the center of the sphere is smaller than the sphere's radius). The above implies that the search point was described ideally and that the required sphere was successfully identified. If I_F is greater than 1, the search point is situated in the space between spheres. The values of I_F calculated with the four discussed methods are presented in Figures 23 and 24. In TNM, the search for spheres is far more effective than in TCM which does not confirm the assumption that the smallest fragments of the bed have a tetrahedral structure in 3D (which is the main reason why TCM should be excluded from further analyses). The relationship between search effectiveness and bed density is also presented in Figure 24 (refer to Fig. 44).

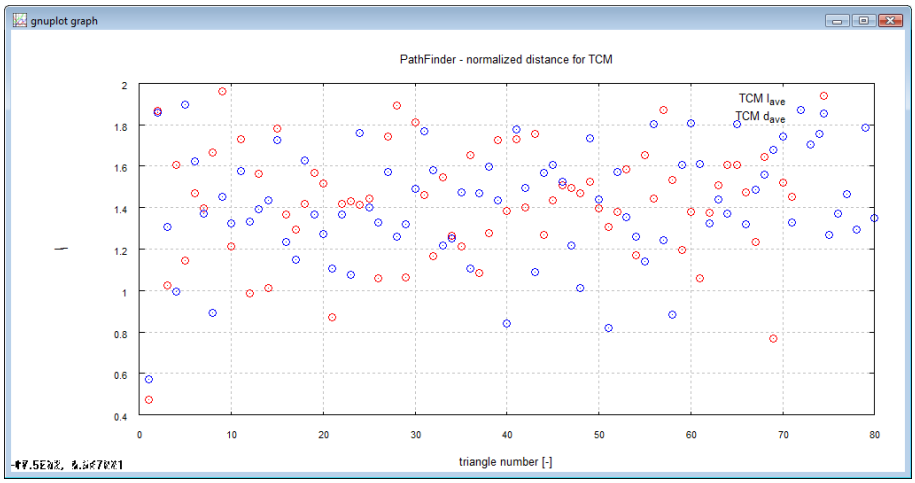

Figure 23. Distribution of I_F in TCM

The average value of the path quality indicator $\overline{I_F}$ was generally used in subsequent analyses. This parameter was applied to evaluate the quality of the sphere search method.

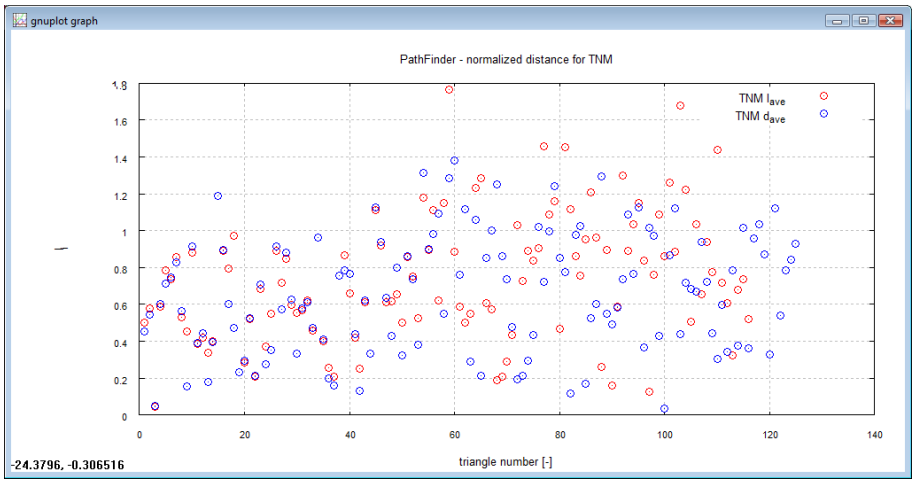

Figure 24. Distribution of I_F in TNM

8. Algorithm sensitivity to changes in specific height

Algorithm sensitivity to changes in the value of C_h was verified in the next stage of the analysis. The results of the calculations are presented in Figures 25-27. TCM was far more sensitive to changes in C_h than TNM, which validates the previous conclusion that the next sphere should be positioned directive "above" the triangle in the direction of its normal. Ultimately, TNM was regarded as a superior model that is better adapted to actual data. It should be noted that all presented values were characterized by similar degree and type of sensitivity. The aim of the simulation was to find the smallest possible value of tortuosity; therefore, the optimal value of the coefficient of specific height was determined at $C_h \approx 0.5$. All of the presented curves achieved minimum values at this value of C_h. At this stage of research, it is difficult to determine whether this is a general characteristic of all beds containing spherical particles, or whether it is a specific feature of the analyzed test bed. The value of C_h up to which the results are minimized should be determined and regarded as correct. This constant can be eliminated in a sensitivity analysis, which generally increases the value of every mathematical model.

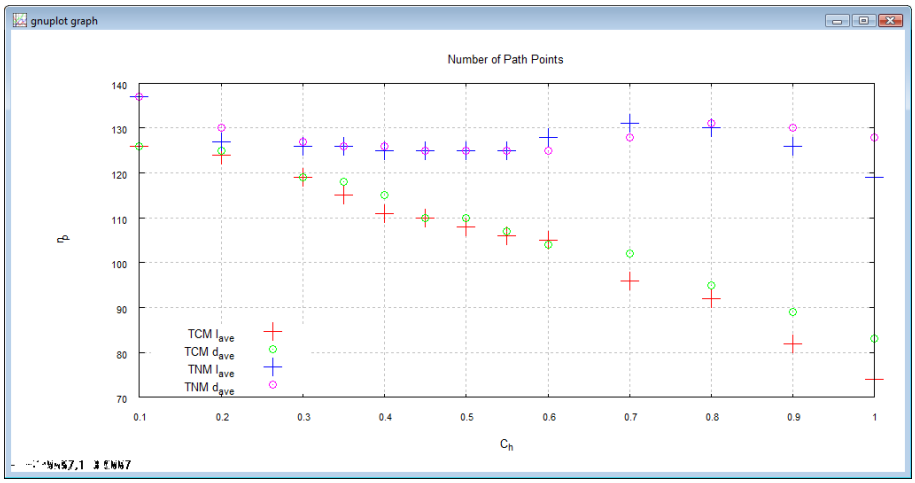

Figure 25. The relationship between coefficient C_h and the number of path points

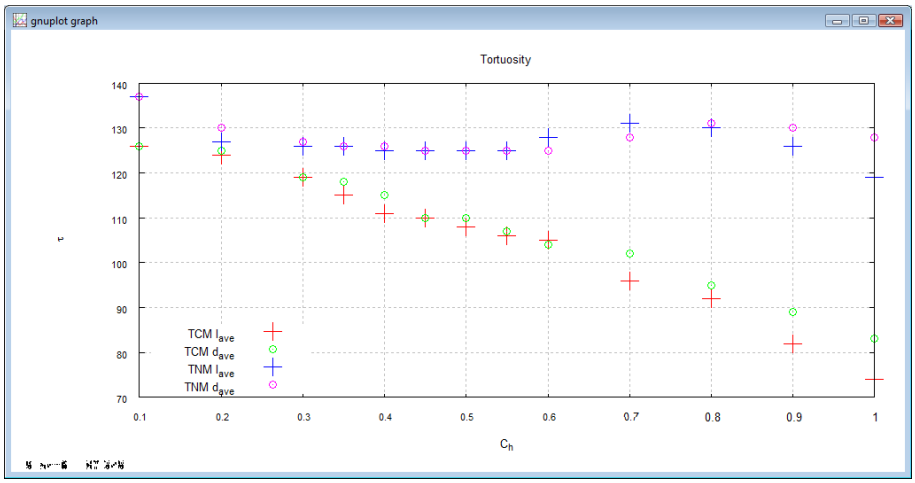

Figure 26. The relationship between coefficient C_h and tortuosity

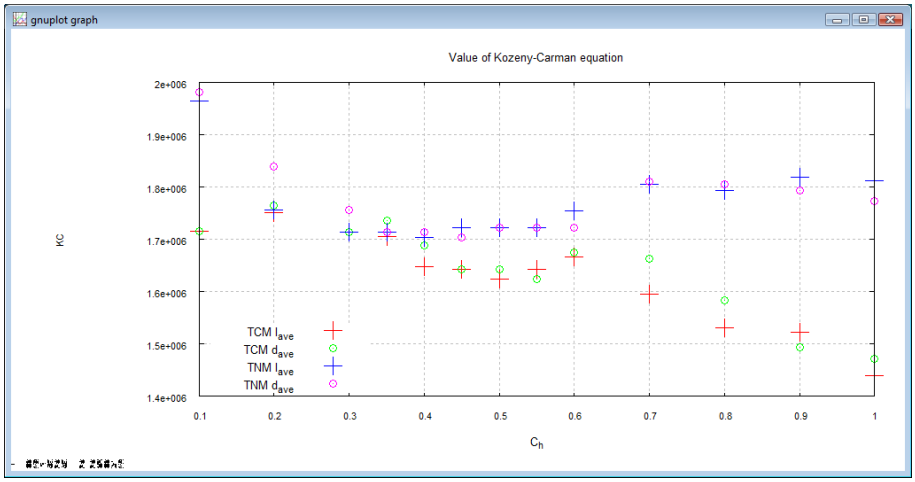

Figure 27. The relationship between coefficient C_h and the value of the Kozeny-Carman equation

The optimal value of coefficient C_h can be explained based on Figure 28. For $C_h = 1$, the search point is positioned relatively high, and a sphere positioned above the search point could be identified as the nearest sphere (test results indicate that this is often the case). The sphere that constitutes the actual tetrahedral vortex is omitted. This is because the search area (red circle) and the triangle area (black line) are separated by unverified space (diagonal lines). When the height of the search point is decreased, this space is eliminated, which increases the probability that the tetrahedral vortex has been correctly identified.

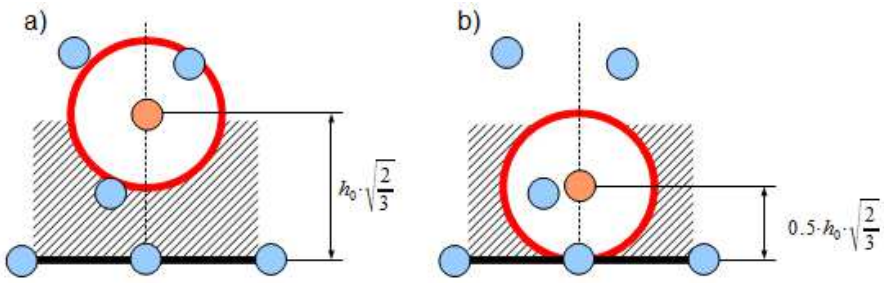

Figure 28. Schematic diagram of the optimal value of coefficient C_h

9. Algorithm sensitivity to changes in the angle of the triangle normal

The model's sensitivity to changes in the angle of the triangle normal was tested in the next stage of the study. The aim of the analysis was to determine whether minor changes in the angle of the normal towards the top layer of the bed would shorten the path and decrease tortuosity (in this case, model TNM would approximate model TCM). The results of the calculations for TNM and $C_h = 1$ are presented in Figures 29-31.

The presented approach was not justified by the laws of physics, and it was ultimately rejected. Successive calculations were performed on the assumption that $C_\gamma = 1$.

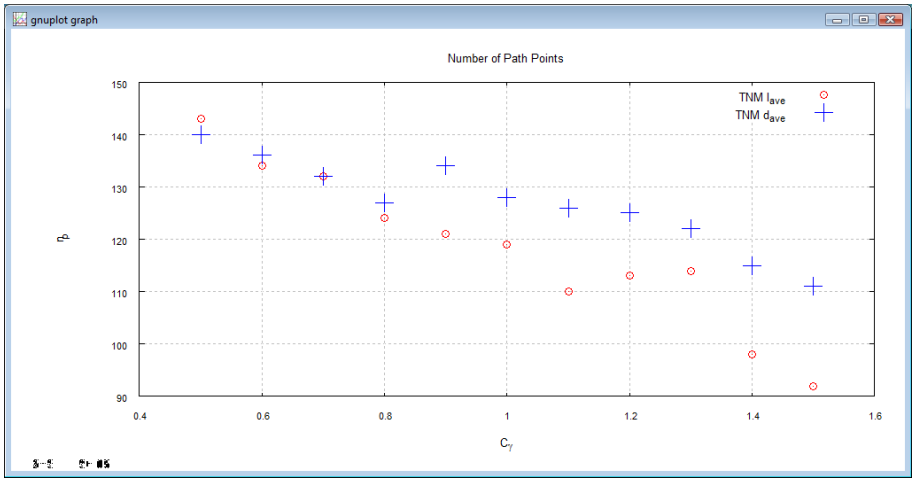

Figure 29. The relationship between coefficient C_γ and the number of path points

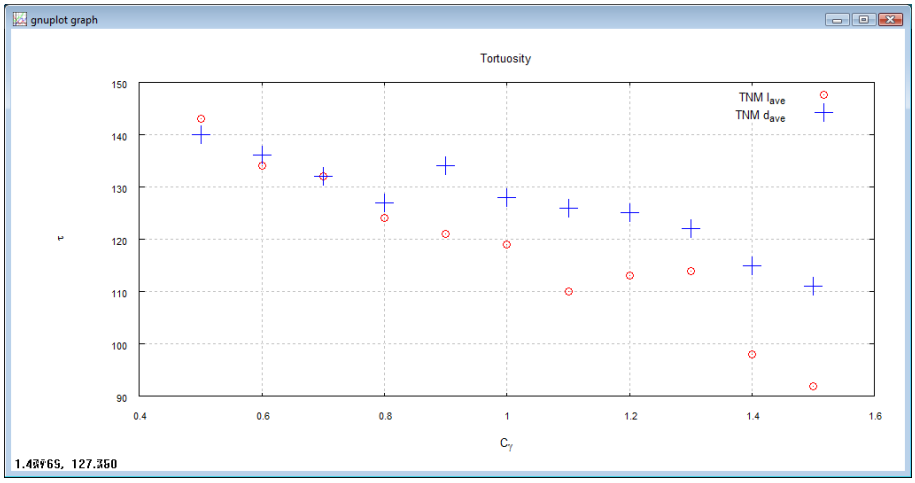

Figure 30. The relationship between coefficient C_γ and tortuosity

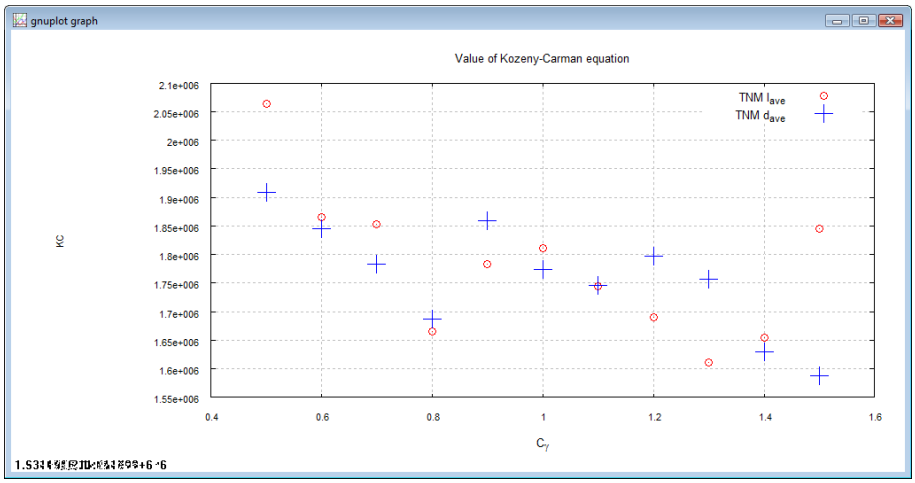

Figure 31. The relationship between coefficient C_γ and the value of the Kozeny-Carman equation

10. Specific height as a function of indicator I_A

If all spheres come have mutual points of contact and form a tetrahedron, the vortex will be separated by a maximum distance from the base. However, the vortex sphere will "fall" in between the spheres forming the base as the distance between those spheres increases. Therefore, if:

$$I_A \to I_A^{cr}, \tag{24}$$

then

$$h \to 0. \tag{25}$$

The above suggests that coefficient C_h with a constant value should not be used; instead, function C_h^f dependent on I_A should be applied:

$$h = h_0 \cdot 2 \cdot C_h^f \cdot \sqrt{2/3}, \tag{26}$$

where

$$\begin{cases} C_h^f(I_A = \overline{I_A}) = h_0 \cdot \sqrt{2/3} \\ C_h^f(I_A = I_A^{cr}) = 0 \end{cases}. \tag{27}$$

The above criteria are met by the following function:

$$C_h^f(I_A) = 1 - \frac{\exp(a \cdot (I_A - b))}{1 + \exp(a \cdot (I_A - b))}. \tag{28}$$

Function $C_h^f(I_A)$ is presented in Figure 32.

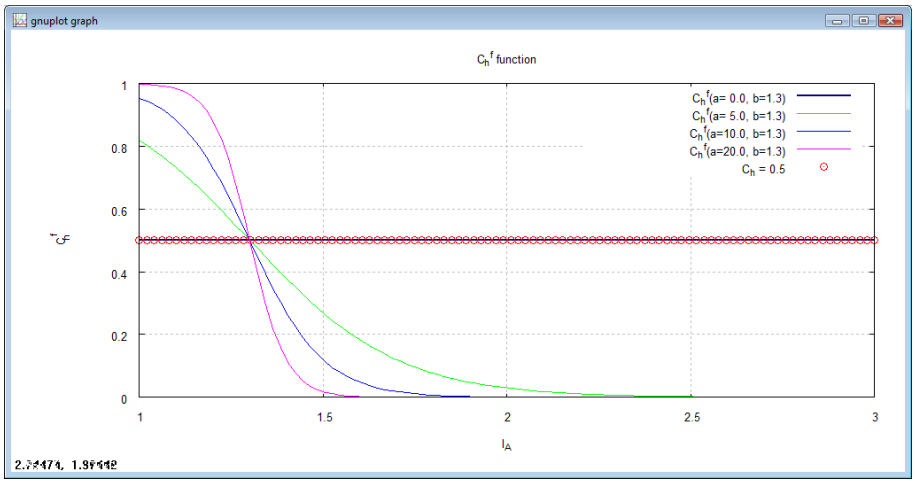

Figure 32. Correcting function C_h^f

Function $C_h^f(I_A)$ has two coefficients. Coefficient a is responsible for the angle of the curve. The exact value of coefficient a can be determined based on the criterion of minimum path length or minimum tortuosity:

Coefficient b should correspond to the average value of $\overline{I_A}$ calculated for the entire bed. As a result, in a "standard" triangle:

$$C_h^f = 0.5 \cdot C_h . \tag{29}$$

The calculated value of $\overline{I_A} \approx 1.3$.

The results of the calculations performed for different values of coefficient a are presented in Figure 33. Function C_h^f significantly decreased tortuosity to the expected level of $3 < a < 9$. The number of spheres identified for the above range of values was identical (Fig. 34). It should also be noted that an increase in the value of a within the above range also increases the average value of I_F which is less than 0.8 for $a = 9$ and is a highly satisfactory result.

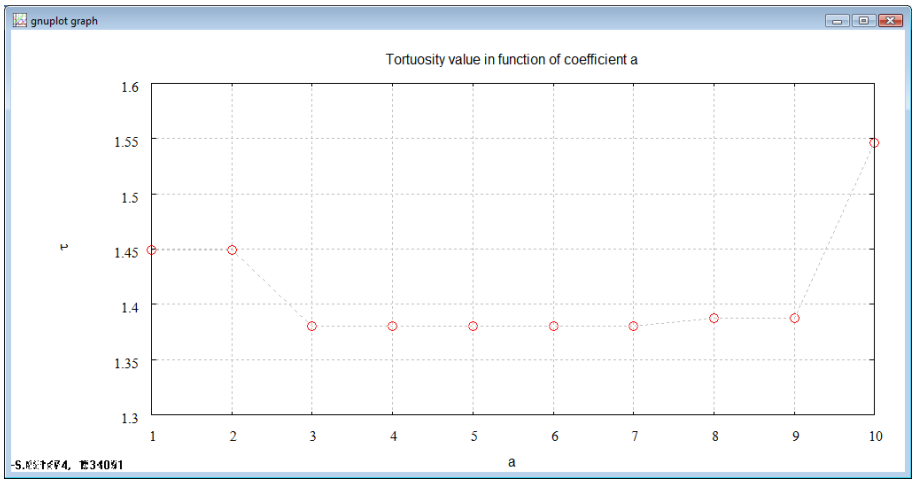

Figure 33. Tortuosity as a function of coefficient a

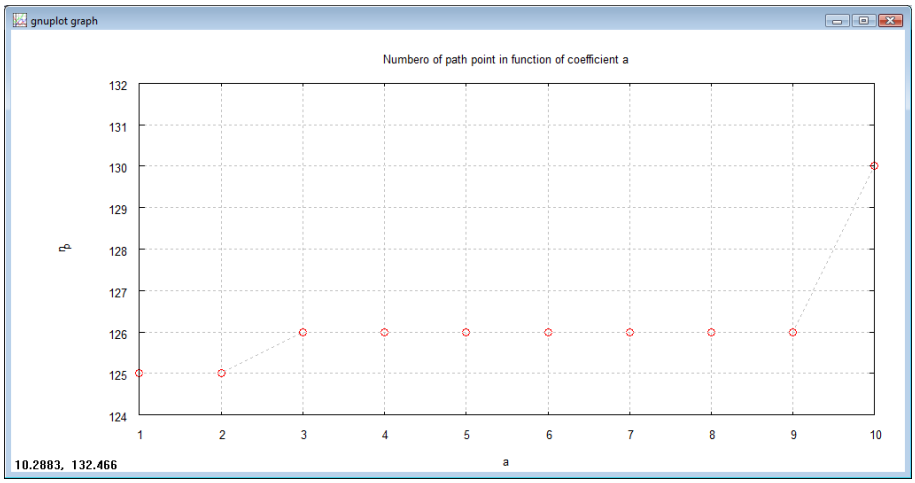

Figure 34. Number of spheres surrounding the path as a function of coefficient a

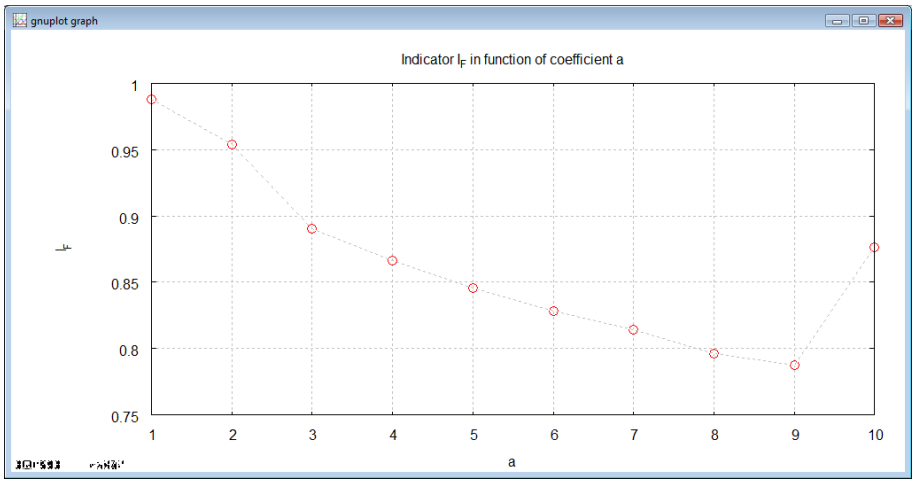

Figure 35. The average value of I_F as a function of coefficient a

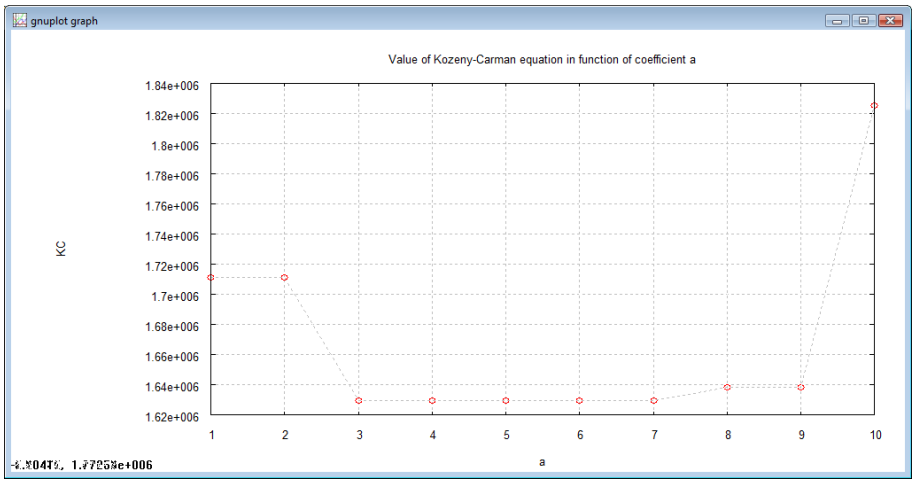

Figure 36. Value of the Kozeny-Carman equation as a function of coefficient a

If $a = 0$, curve $C_h^f(I_A)$ becomes a straight horizontal line and the value of the function is always 0.5. This implies that the results for $C_h = 0.5$ and $C_h^f(a = 0, b = 1.3)$ should be identical. The above was fully validated by the performed tests.

11. Selection of the starting point and determination of average tortuosity

In previous analyses, the starting point was always positioned centrally inside the bed. This section will aim to determine the magnitude of differences in the results if various starting points are adopted. In addition to the central point, four points were positioned at mid-distance between the center and the container wall on negative and positive X-axes and tortuosity (refer to Fig. 37 and 38). The results are presented in Tables 4 and 5.

Table 4. The results of the calculations for 5 starting points in TNM, $h_0 = l_{ave}$, $C_h = 0.5$

Position	n_p	I_F	τ	KC
+x	125	1.02503	1.44958	1721930.07710
-x	130	1.02578	1.51340	1797734.02860
+y	129	1.01683	1.51845	1803737.05410
-y	129	1.00480	1.48376	1762528.26340
0	128	0.99598	1.45603	1729589.67280
Average	**128.20**	**1.01368**	**1.48424**	**1763103.81920**

Table 5. The results of the calculations for 5 starting points in TNM, $h_0 = l_{ave}$, $C_h^f (a = 9.0, b = 1.3)$

Position	n_p	I_F	τ	KC
+x	126	0.81438	1.38040	1639752.25080
-x	130	0.83741	1.50890	1792388.36080
+y	133	0.86315	1.54804	1838881.49840
-y	127	0.91392	1.40253	1666040.01190
0	132	0.92779	1.62483	1930101.19820
Average	**129.60**	**0.87133**	**1.49294**	**1773432.66402**

The selection of different starting points led to variations in path length. The above indicates that calculations should be performed at numerous points inside the bed and that the results should be averaged. The number of such calculations remains debatable. Five points were adopted in successive calculations.

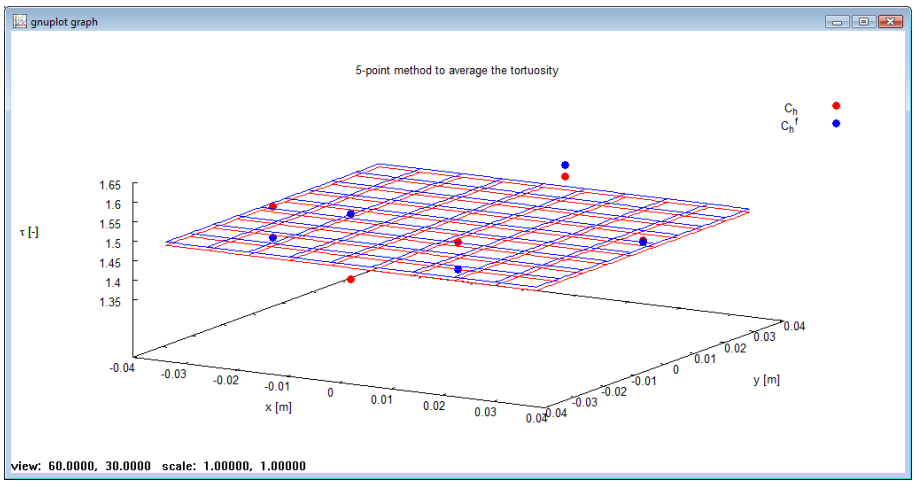

Figure 37. Tortuosity values in TNM, $h_0 = l_{ave}$

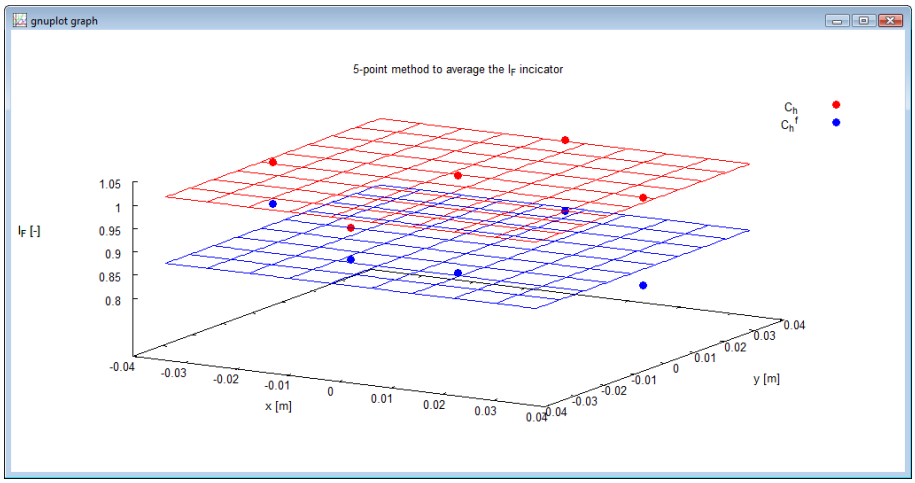

Figure 38. Values of indicator I_F in TNM, $h_0 = l_{ave}$, $C_h^f \left(a = 9.0, b = 1.3\right)$

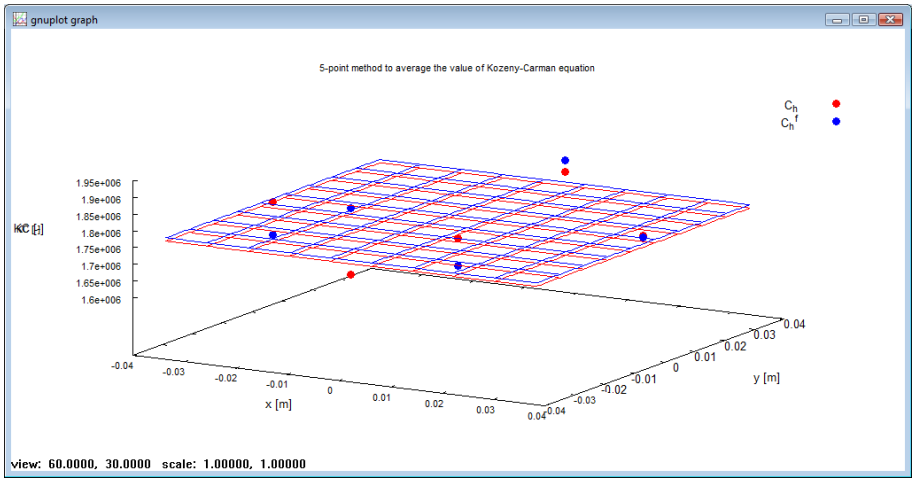

Figure 39. Values of the Kozeny-Carman equation in TNM, $h_0 = l_{ave}$, $C_h^f \left(a = 9.0, b = 1.3\right)$

The paths generated by both methods are visualized in Figures 40 and 41.

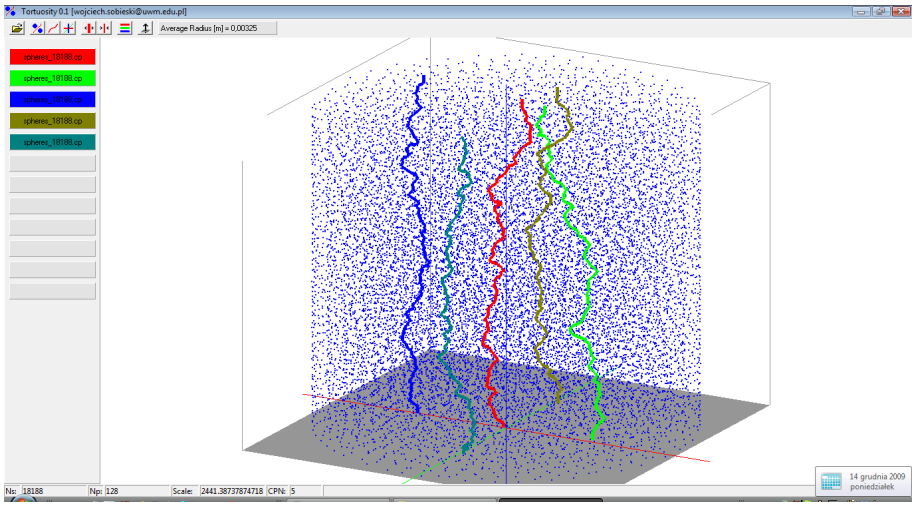

Figure 40. Visualization of paths in TNM, $h_0 = l_{ave}$, $C_h = 0.5$

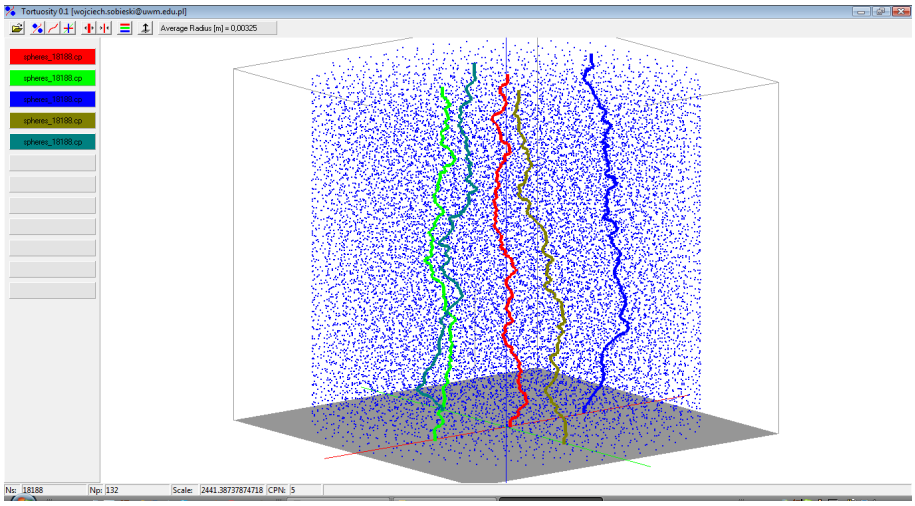

Figure 41. Visualization of paths in TNM, $h_0 = l_{ave}$, $C_h^f \left(a = 9.0, b = 1.3\right)$

12. Determination of bed porosity

The developed program generates information about the number and size of all spheres in the bed and supports the determination of bed porosity. The standard formula for calculating porosity was used:

$$e = 1 - \frac{V_s}{V_{bin}}, \tag{30}$$

where V_s is the combined volume of all particles [m^3]:

$$V_s = \sum_{i=1}^{n_s} \frac{\pi \cdot d_i^3}{6}, \tag{31}$$

and V_{bin} is container volume [m^3]

$$V_{bin} = \pi \cdot \left(\frac{D_{bin}}{2}\right)^2 \cdot \left(z(n_s) - z(1) + d_{ave}\right). \tag{32}$$

Symbol D_{bin} denotes the diameter of the container with the test bed [m].

Porosity was determined at $e = 0.4265$, and it is somewhat higher than the experimentally determined value of $e_{\exp} = 0.408$. This minor difference can probably be attributed to the method of calculating container volume, specifically its height. Container height was determined as the difference between the Z-coordinate of the topmost and the bottommost sphere in the bed. The sphere radius multiplied by two was added to the resulting value. The presented formula contains a logical error because unlike the bottom layer of the bed, the top layer is not ideally flat. As a result, a small empty space between the last sphere layer and the plane at height $z(n_s) - z(1) + \frac{d_{ave}}{2}$ was included in the calculations of porosity. In Figure 42, this empty area is

visible under the grey line representing the indicated plane. The formula for calculating container volume (bed volume) should be adjusted to account for the above issue.

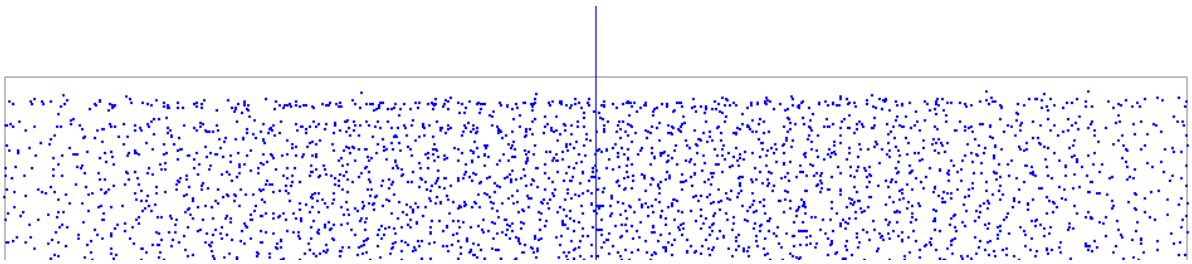

Figure 42. Visualization of the top surface of the container.

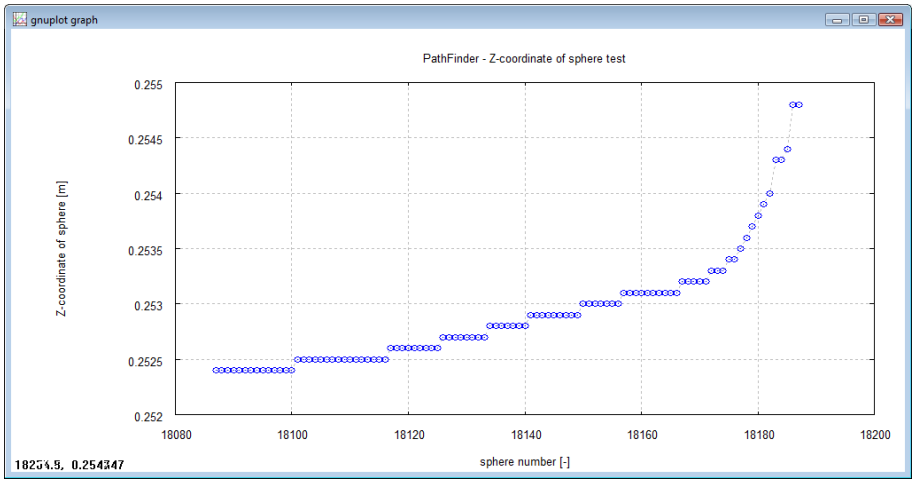

Figure 43. The coordinates of 100 topmost spheres.

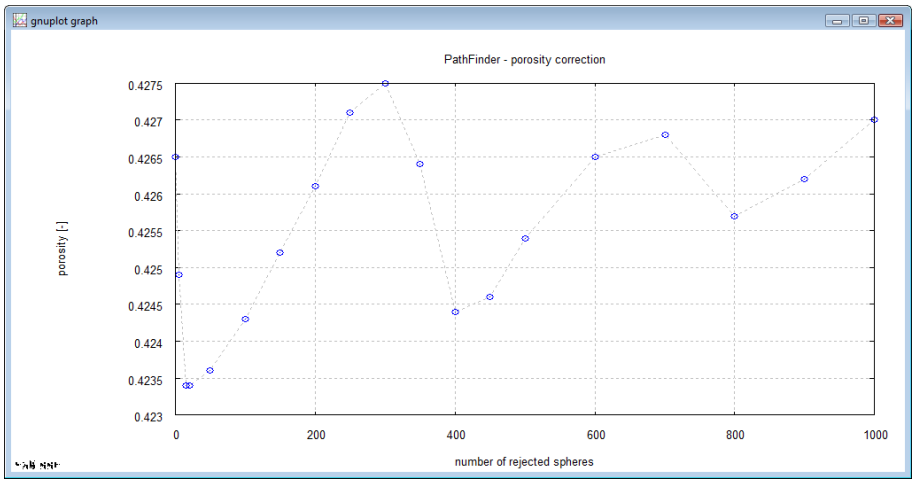

Figure 44. Changes in porosity as a function of the number of eliminated spheres

The formula was adjusted by verifying the Z-coordinates of 100 topmost spheres. The result is shown in Figure 43. A certain number of topmost spheres were positioned on other spheres in a manner that significantly increased their coordinates. The results presented in Figure 43 indicate that a certain number of topmost spheres should be rejected.

Formulas (31) and (32) were modified as follows:

$$V_{bin} = \pi \cdot \left(\frac{D_{bin}}{2} \right)^2 \cdot \left(z(n_s - n_{cor}) - z(1) + d_{ave} \right). \tag{33}$$

and

$$V_s = \sum_{i=1}^{n_s - n_{cor}} \frac{\pi \cdot d_i^3}{6}, \tag{34}$$

where n_{cor} is the number of rejected spheres.

The changes in porosity induced by the number of eliminated spheres are presented in Figure 44. The combined volume of all spheres and bed height (container volume) were corrected in every case. Changes in porosity were smaller than expected, and the result was not significantly modified by the introduced correction. The above indicates that the differences between the porosity calculated in the program and that simulated by PFC3D have different causes. It is possible that the changes resulting from the elimination of a certain number of topmost spheres are far smaller than the changes in bed porosity at that height. The results presented in the next section indicate that changes in bed porosity are considerable.

13. Influence of porosity on path quality

On several occasions, the operation of the program deteriorated beginning from a certain height, approximately 2/3 of bed height. The size of the triangles was disrupted above that level, and the correction described in point 6 (refer to Fig. 20) had to be applied. The observed disruption was probably caused by changes in bed porosity. Figure 44 presents the distribution of porosity, calculated in PVC3D, for five points in the bed – in the center and at four points on the semi-axes of the coordinate system at mid-distance between the center and the wall of the container. These points were marked "+x", "-x", "+y" and "-y", respectively. Figure 45 indicates that the algorithm deteriorated above the bed height where minimum porosity was observed.

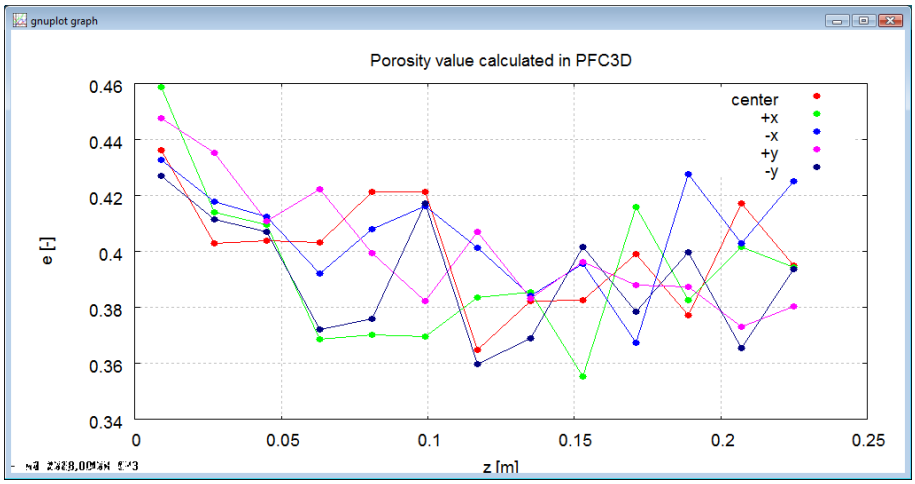

Figure 45. Distribution of porosity in PFC3D

14. Sensitivity analysis of the Ergun model

The developed program calculates the terms of equations supporting the determination of the drop in pressure during fluid flow through a porous bed. The program presently supports two models: the Ergun model and the Kozeny-Carman model. The Ergun model (1952) has the following form:

$$-\frac{dp}{dL} = 150 \cdot \frac{(1-e)^2}{e^3 \cdot d_{ave}^2} \cdot \mu \cdot v + 1.75 \cdot \frac{(1-e)}{e^3 \cdot d_{ave}} \cdot \rho \cdot v^2, \tag{35}$$

where: v – filtration rate [m/s], ρ – fluid density [kg/m^3], μ – dynamic viscosity [kg/(m·s)]. The Ergun equation can also be written as:

$$-\frac{dp}{dL} = A_E \cdot \mu \cdot v + B_E \cdot \rho \cdot v^2 \tag{36}$$

where

$$A_E = 150 \cdot \frac{(1-e)^2}{e^3 \cdot d_{ave}^2}, \tag{37}$$

and

$$B_E = 1.75 \cdot \frac{(1-e)}{e^3 \cdot d_{ave}}. \tag{38}$$

This form of the Ergun equation corresponds to the Forchheimer law.

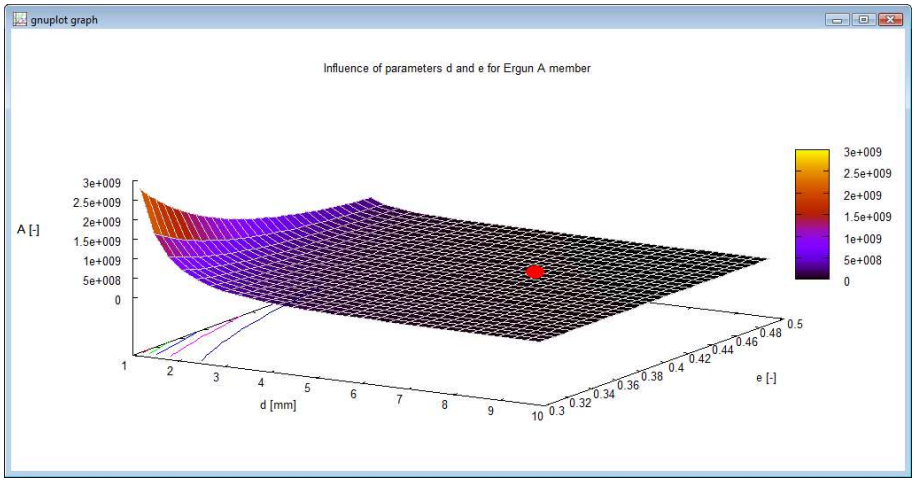

Figure 46. The influence of porosity and average sphere diameter on the value of term A in the Ergun model

The Ergun equation is generally a function of two geometric parameters, and the model's sensitivity to changes in these parameters was analyzed. A designated program was developed for this purpose to illustrate the spatial distribution of terms A and B in the Ergun model. The results are presented in Figures 46 and 47. The positions corresponding to the values from the test bed are marked with red dots. Both points are positioned in a flat area, which indicates that

inaccurate values of these parameters should not significantly influence the drop in pressure computed in that model. The distribution of pressure drop values for this formula is presented in Figure 48.

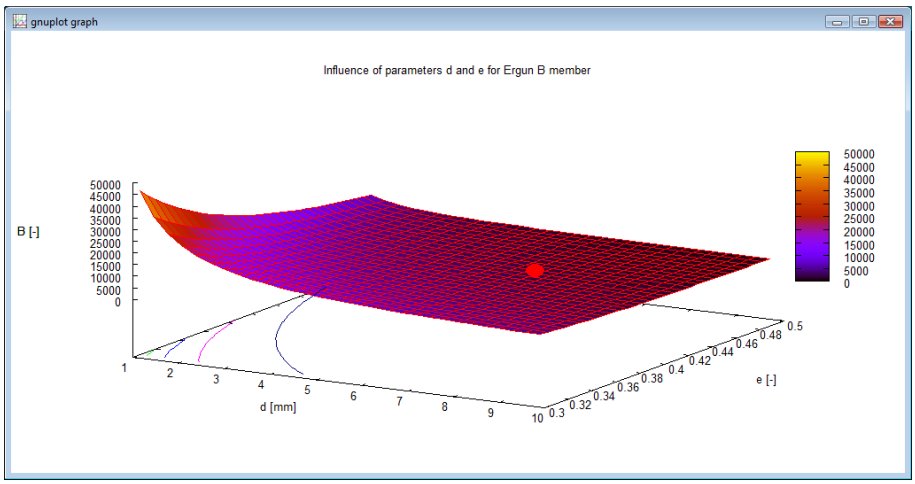

Figure 47. The influence of porosity and average sphere diameter on the value of term B in the Ergun model

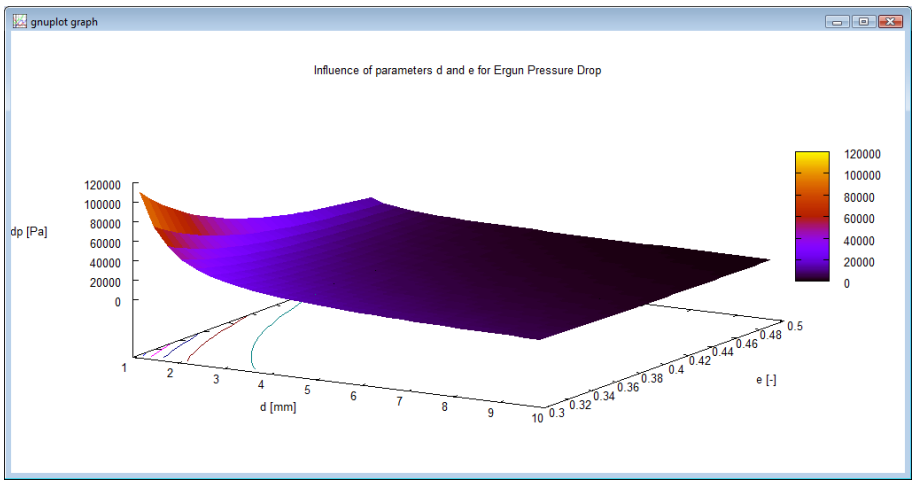

Figure 48. The influence of porosity and average sphere diameter on the pressure drop described by the Ergun equation

15. Sensitivity analysis of the Kozeny-Carman model

The Kozeny-Carman equation was the second model to be implemented in the program. This model was developed for well sorted sand (a medium with a finer structure than the presented test bed), and it has the following form:

$$-\frac{dp}{dL} = c \cdot \tau \cdot S_0^2 \cdot \frac{(1-e)^2}{e^3} \cdot \mu \cdot v, \tag{39}$$

where: c - constant, τ - tortuosity, S_0 - surface area characteristic of a porous body. The Kozeny-Carman equation can be written as follows:

$$-\frac{dp}{dL} = A_{KC} \cdot \mu \cdot v, \tag{40}$$

where

$$A_{KC} = c \cdot \tau \cdot S_0^2 \cdot \frac{(1-e)^2}{e^3}. \tag{41}$$

Tortuosity was defined at the beginning of the study. Specific surface area is determined with the use of the below formula:

$$S_0 = \frac{S_s}{V_{bin}}, \tag{42}$$

where: S_s - external surface area of a porous body [m^2], V_{bin} - bed porosity [m^3]. Container volume was calculated based on formula (33), and sphere area was calculated according to formula (43):

$$S_s = \sum_{i=1}^{n_s - n_{cor}} \pi \cdot d_i^2 . \tag{43}$$

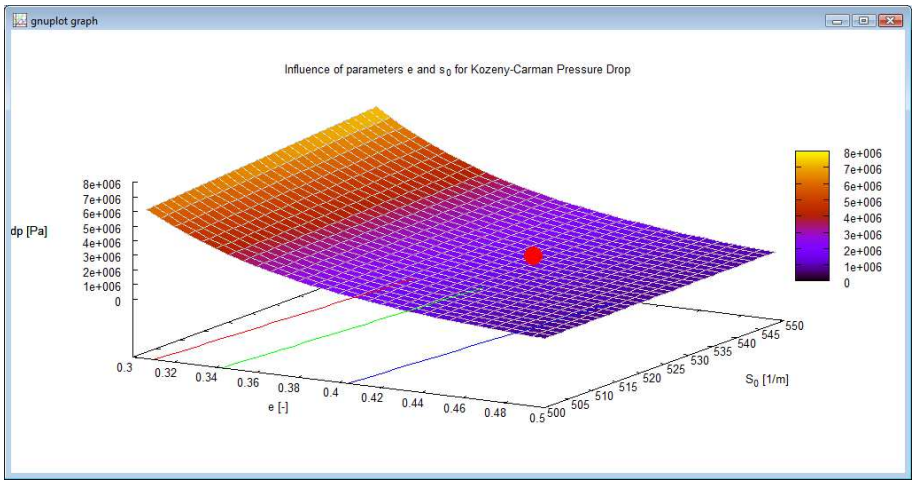

Figure 49. The influence of porosity and surface area characteristic of a porous body on the pressure drop described by the Kozeny-Carman equation.

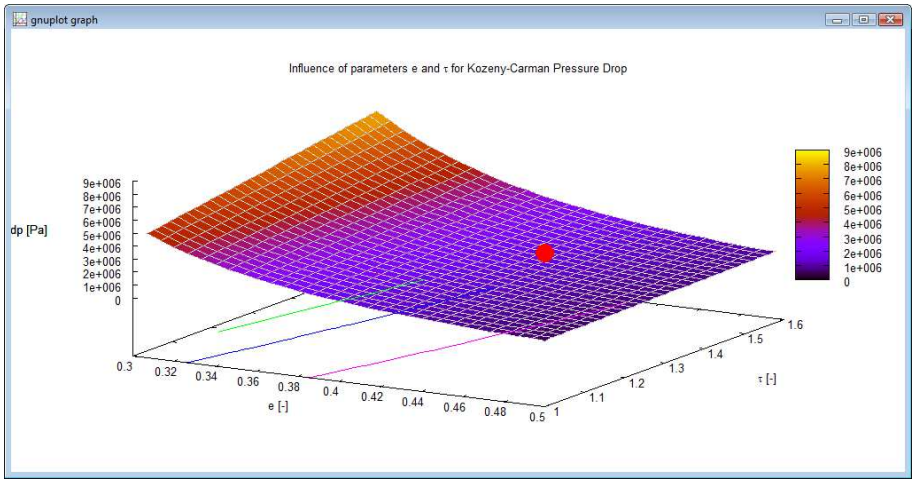

Figure 50. The influence of porosity and tortuosity on the pressure drop described by the Kozeny-Carman equation

The results of the analysis are presented in Figures 49 and 50 which indicate that the Kozeny-Carman model is much more sensitive to changes in geometric parameters. This outcome was expected because the formula has a more complex form, which is strongly influenced by the geometry of the tested medium.

16. Determination of optimal model parameters

Based on the results of the conducted tests, the optimal configuration of parameters was presented in Table 6. For those parameters, tortuosity was determined at $\tau = 1.49294$ (identical to that presented in Table 5; the decrease in coefficient value from 0.9 to 0.7 did not affect the result).

Table 6. Optimal configuration of the modeled parameters

Method of calculating search point coordinates	TNM
Method of calculating height h_0	l_{ave}
Method of correcting height h_0	$C_h^f(a = 7.0, b = 1.3)$
Correction factor C_γ	1.0
Number of spheres eliminated from the top layer of the bed n_{cor}	5
Triangle correction for $I_A = I_A^{cr}$	YES
Five-point averaging	YES

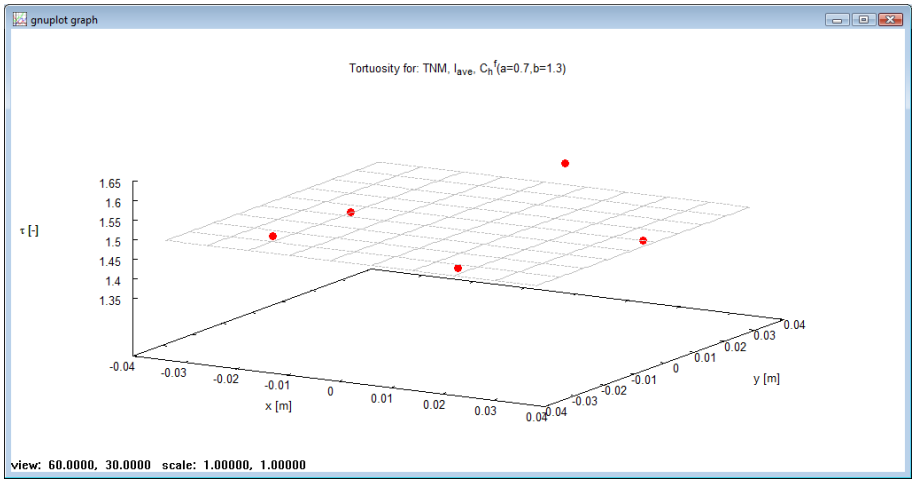

Figure 51. Tortuosity values calculated for the parameters in Table 6.

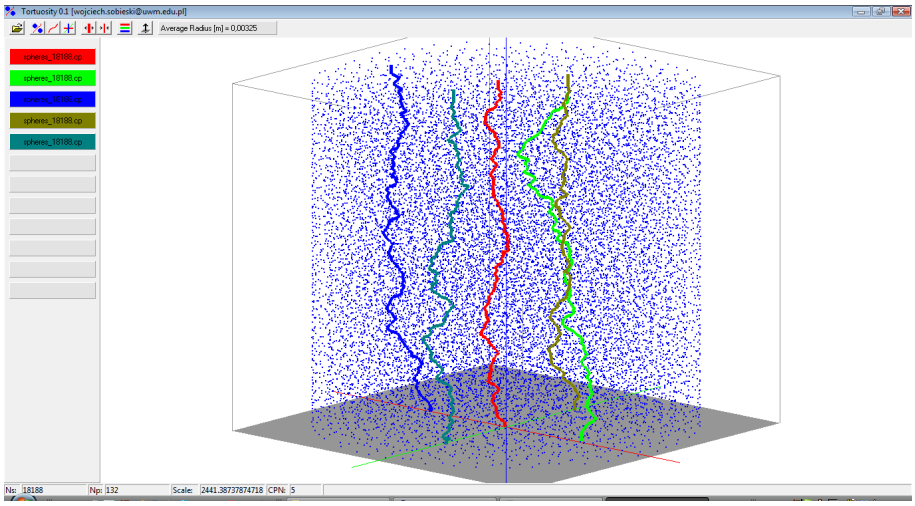

Figure 52. Visualization of paths for the parameters in Table 6

Path length and, consequently, tortuosity (Fig. 51), calculated for the values in Table 6 seem to be excessive. The quality of the calculations could be improved with path smoothing algorithms, and this problem will be discussed in the next section. The optimal method should include a parameter for controlling the degree of smoothing.

17. Path smoothing

Selected segments of the path form a polygonal chain with acute angles (Fig. 53). Path smoothing algorithms could be applied to model the path more realistically. This problem can be analyzed separately from the presented solutions. The only required information for path smoothing is the position of each point in space and the number of points. Therefore, path smoothing is not directly related to the methods described in this article.

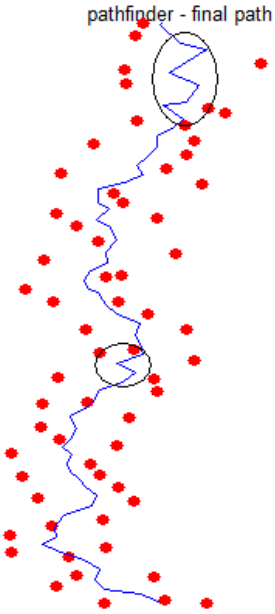

Figure 53. Visualization of a path segment in TNM, $h_0 = l_{ave}$, $C_h^f\left(a = 0.5, b = 1.3\right)$

The following options can be considered in the path smoothing process:

- elimination of "mutants" (Fig. 54a) – selected path points are unnaturally distant from the course of the path. An example is shown in Figure 53 (more or less in the center). This point could be disregarded in the calculations. The selection of a search algorithm and the criteria for selecting such objects are open to debate.
- rounding off path breakdown points (Fig. 54b) – some segments of the path are relatively smooth, but frequent and rapid changes in direction are observed in other areas (marked at the top of the path in Fig. 53). The path could be approximated with a curve, and its length could be calculated after this transformation. The path should be somewhat shorter, which is generally desirable. This is probably the most interesting avenue of future research.
- incorporation of tetrahedral centers (Fig. 54c) – tetrahedral centers could be incorporated into the path (after path smoothing by approximation). The resulting path could approximate reality more effectively.

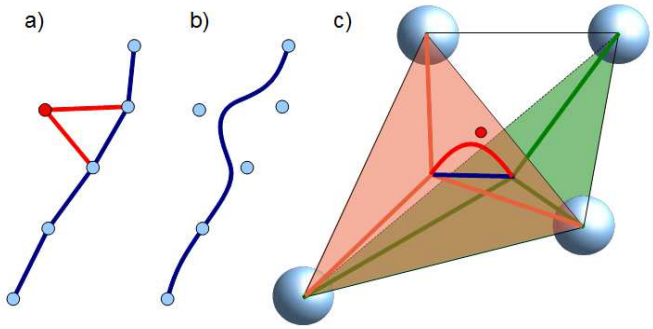

Figure 54. Visualization of path smoothing methods

The use of Bezier curves could also be considered. A previously designed program for plotting Bezier curves in space is presented in Figure 55.

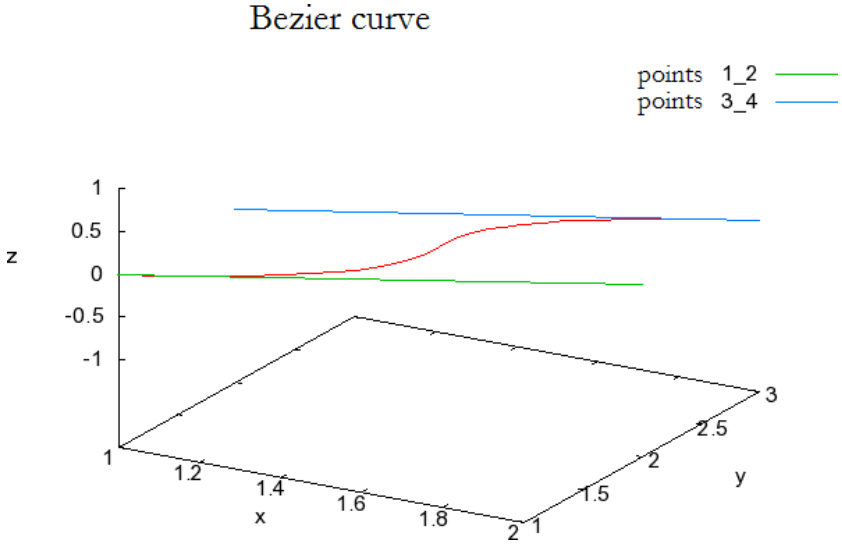

Figure 55. Path smoothing with Bezier curves (previously designed program)

18. Path smoothing methods

The following conclusions can be drawn from an analysis of path smoothing methods:

- The elimination of selected path points is not logically justified. A control parameter is required to determine which path points should be rejected. This method was ultimately abandoned.
- The method that relies on Bezier curves for 3D approximation also seems to have numerous defects:
  - it is difficult to implement,
  - two auxiliary points are required to control the shape of the curve, which generates additional problems relating to point selection,
  - successive algorithms are required for integrating the path and calculating its length.

The above problems indicate that path length should be adjusted locally, depending on the angle between the adjoining path segments. The angle was calculated based on the law of cosines:

$$\cos\varpi_i = \frac{a_i^2 + b_i^2 - c_i^2}{2 \cdot a_i \cdot b_i}, \tag{44}$$

where i is the number of the current path point, and variables a_i, b_i and c_i denote the side lengths of a triangle formed by three adjacent path points, as demonstrated in Figure 56a.

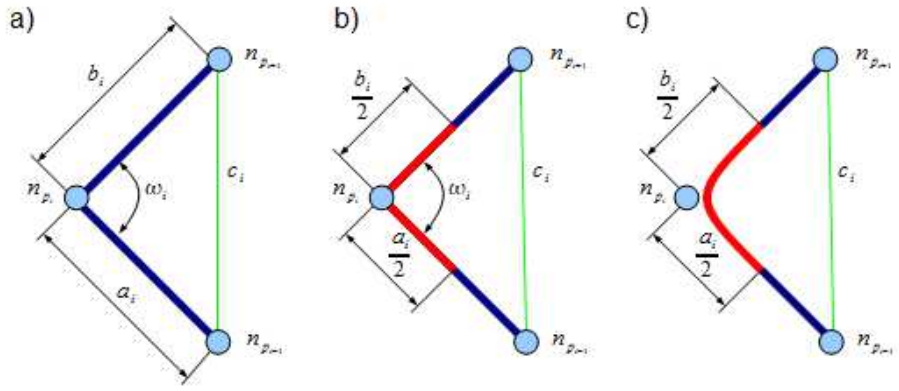

Figure 56. Schematic diagram of the path smoothing method

The value of angle ϖ (in degrees) for the case described in section 16 is presented in Figure 57. In the discussed case, the average value of angle ϖ was 140.13 degrees.

The path was divided into segments with length $\frac{a_i}{2} + \frac{b_i}{2}$. The length of successive segments would decrease, subject to the value of angle ϖ_i. An additional correcting function was introduced for this purpose:

$$\varpi_i^{cor} = \frac{\exp(\varpi_a \cdot (\varpi_i - \varpi_b))}{1 + \exp(\varpi_a \cdot (\varpi_i - \varpi_b))}. \tag{45}$$

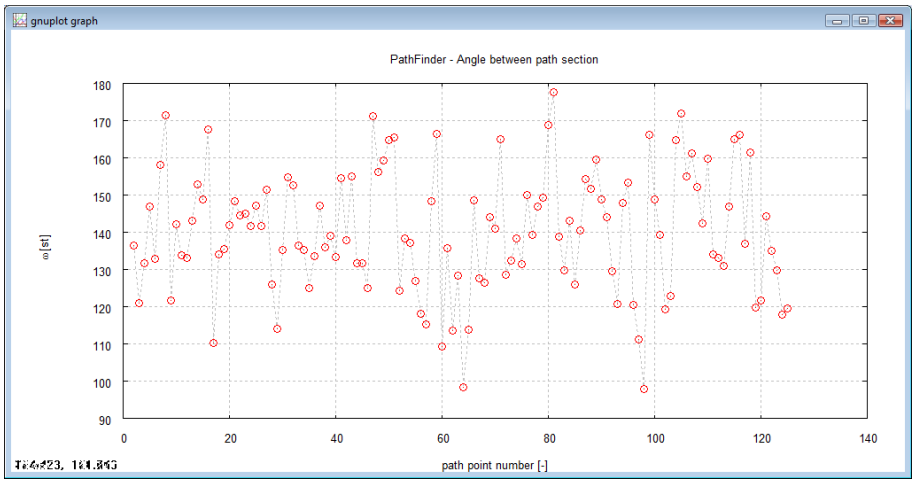

Figure 57. Values of the angle between successive segments of the path for the case analyzed in section 16.

The function is presented in Figure 58. The correction factor equals 0 for $\varpi = 180$. The correction factor also approximates zero for smaller angles, and it increases very slowly. The rate of increase in the value of ϖ^{cor}_i is determined by coefficient ϖ_a.

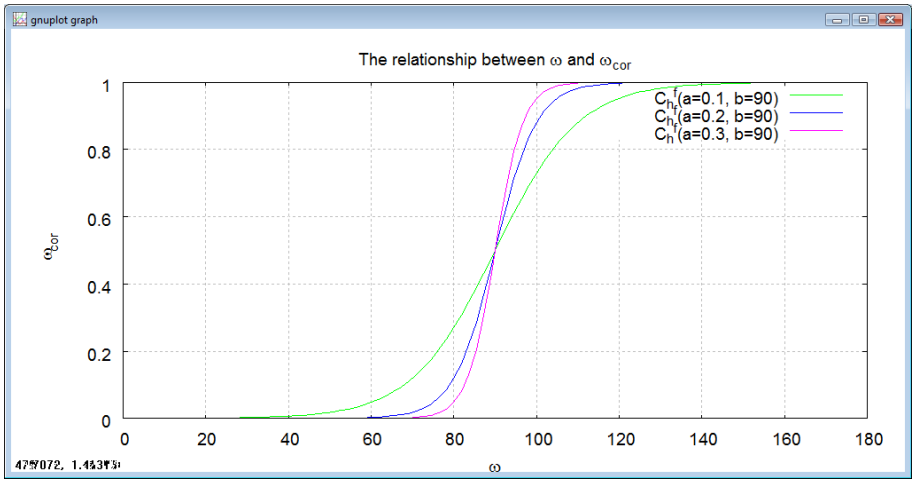

Figure 58. The relationship between angle ϖ and the correction factor.

After the correction, the formula for calculating path length will have the following form:

$$L_p^{cor} = \frac{1}{2}\left(a_1 + b_{n_p}\right) + \sum_{2}^{n_p-1} \frac{1}{2}\left(a_i + b_i\right) \cdot \varpi_i^{cor} \tag{46}$$

The first term of the equation accounts for the first half of the first path segment and the second half of the last path segment. The sum of halves of the segments adjacent to the current path point (red segments in Fig. 56b), multiplied by the correction factor, is shown under the summation sign. This value is generally less than 1; therefore, the resulting value is somewhat smaller than the sum of $\frac{a_i}{2} + \frac{b_i}{2}$ (and is always determined by the angle between segments). A decrease in the length of this path segment is equivalent to path rounding in its current point, as demonstrated in Figure 56c.

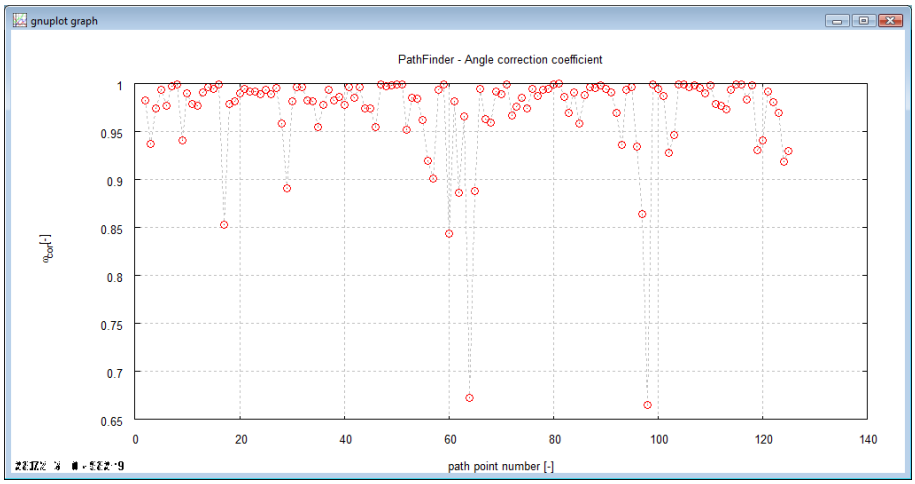

Figure 59. The value of the correction factor for the case analyzed in section 16

The value of the correction factor for the case analyzed in section 16 is presented in Figure 58. Coefficient $\varpi_a = 0.087$ (selected experimentally), whereas $\varpi_9 = 90$ (constant value in the model, equivalent to 90°). The value of tortuosity for the above parameters is $\tau^{cor} = 1.3$.

A comparison of the data in Figures 57 and 59 clearly indicates that path points with the smallest angle are most extensively corrected.

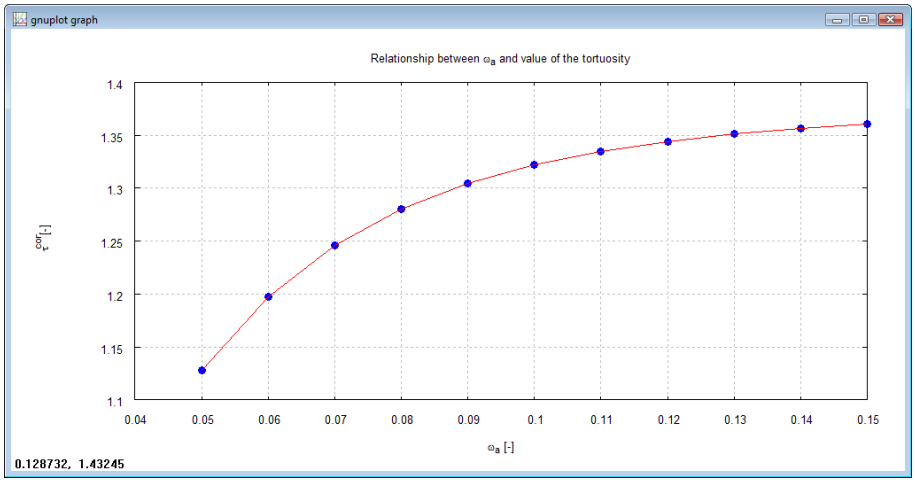

Figure 60. The relationship between correction factor ϖ_a and the value of τ^{cor} for the case analyzed in section 16

The relationship between correction factor ϖ_a and the value of τ^{cor} for the case analyzed in section 16 is presented in Figure 60. The correction factor for the discussed test bed should range from 0.08 to 0.09.

19. Determination of path length by 2D image analysis

The applicability of 2D image analysis methods for determining path length was analyzed in this part of the article. The authors relied on study [6], where the distribution of particles was visualized with the use of X-rays (Fig. 1c in the cited study). As expected, the calculation of path length in a 2D image did not approximate reality, and the result was considerably underestimated.

Three paths with a length of 138.83, 136.55 and 136.51 (units are not relevant) and tortuosity of 1.173, 1.135 and 1.134, respectively, are presented in Figure 61. These values are too low, which can be attributed to the fact that 2D images do not account for changes in path depth; therefore, the path is shortened.

In the presented calculations, the percentage increase in length in plane XZ for the three performed measurements was determined at 8.3%, 6.52% and 6.49%. If we assume that a similar increase will take place in plane YZ, path length should be doubled to account for the percentage value. When this assumption is taken into account, path lengths will increase to 149.47, 144.92 and 144.83. The respective tortuosity values for path length L_p will reach 1.360, 1.278 and 1.276. Average tortuosity will be 1.3, and it corresponds to $\tau^{cor} = 1.3$ calculated in the previous section.

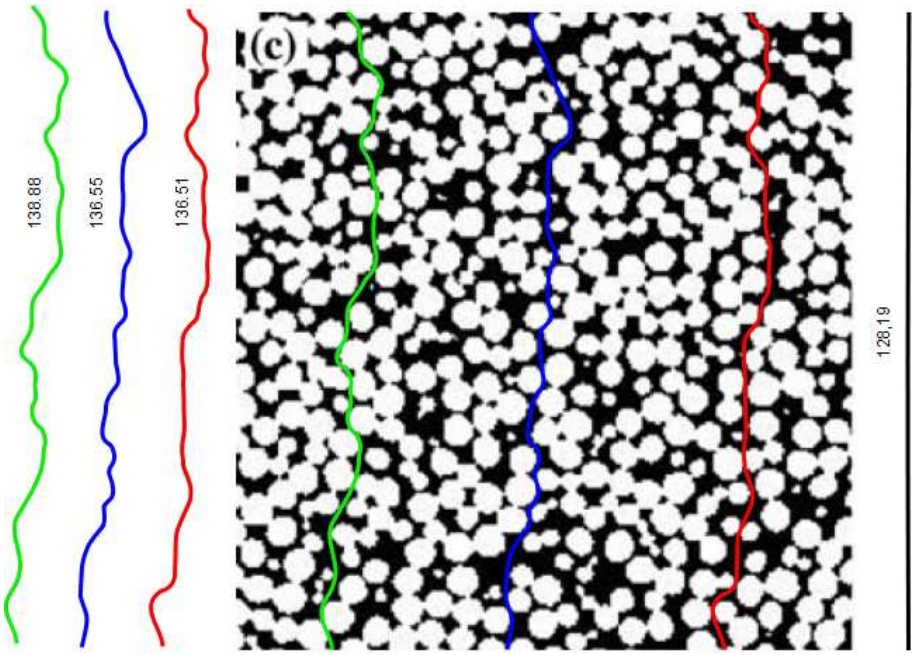

Figure 61. Calculation of path length with the use of an image analysis method described in [6]

20. Tortuosity values in the literature

The importance of tortuosity has been emphasized in the literature, but very few studies make a reference to the actual values of this parameter. The above can be attributed to the fact that tortuosity is very difficult to calculate and measure.

21. Conclusions

The results of this study support the formulation of the following conclusions:

- The value of tortuosity can be calculated if the relevant data, including the number of spheres in the bed, sphere diameter and position, are available.
- The TNM model described in this study appears to be logically and mathematically cohesive, and the generated results can be useful in practice. It appears that the remaining methods should be rejected.
- Path length and, consequently, tortuosity appear somewhat excessive.

- The path can be smoothed, but the proposed method requires a constant smoothing parameter, which is a certain disadvantage. The constant seems to be independent of the selected smoothing method because the tortuosity of fluid flowing in the same path cannot be unambiguously determined.
- The actual value of tortuosity in the test bed or in a medium with a very similar structure should be determined experimentally to generate more accurate results.

R E F E R E N C E S

1. Fourie W., Said R., Young P., Barnes D. L.: The Simulation of Pore Scale Fluid Flow with Real World Geometries Obtained from X-Ray Computed Tomography. COMSOL Conference Boston, United States, 2007.
2. Gas Flow in Porous Media – Turbulence or Thermodynamics
3. Liu Ch., Zhang Q., Chen Y.: PFC3D Simulations of Lateral Pressures in Model Bin. ASABE International Meeting, paper number 083340. Rhode Island, USA 2008.
4. Liu Ch., Zhang Q., Chen Y.: PFC3D Simulations of Vibration Characteristisc of Bulk Solids in Storage Bins. ASABE International Meeting, paper number 083339. Rhode Island, USA 2008.
5. Neithalath N., Weiss J., Olek J.: Predicting the Permeability of Pervious Concrete (Enhanced Porosity Concrete) from Non-Destructive Electrical Measurements. Available at https://fp.auburn.edu/heinmic/perviousconcrete/Porosity.pdf (October, 5, 2009)
6. Neethirajan S., Jayas D. S.: Analysis of Pore Network in Three-dimensional (3D) Grain Bulks Using X-ray CT Images. Transport in Porous Media, Vol 73(2008), pp.319–332
7. DOI 10.1007/s11242-007-9172-x
8. Neethirajan S., Karunakaran C., Jayas D. S., White N. D. G.: X-ray Computed Tomography Image Analysis to explain the Airflow Resistance Differences in Grain Bulks. Biosystems Engineering (2006) 94 (4), pp. 545–555.
9. Sawicki J., Szpakowski W., Wejnerowska K., Wołoszyn E., Zima P. : Laboratorium z Mechaniki Płynów, Gdańsk, 2004.
10. Sobieski W., Trykozko A.: Darcy and Forchheimer laws in experimental and simulation studies of flow through porous media. Transport In Porous Media (submitted).
11. Barrande M., Bouchet R., Denoyel R.: Tortuosity of Porous Particles. Anal. Chem. 2007, 79, pp. 9115-9121.