PathFinder

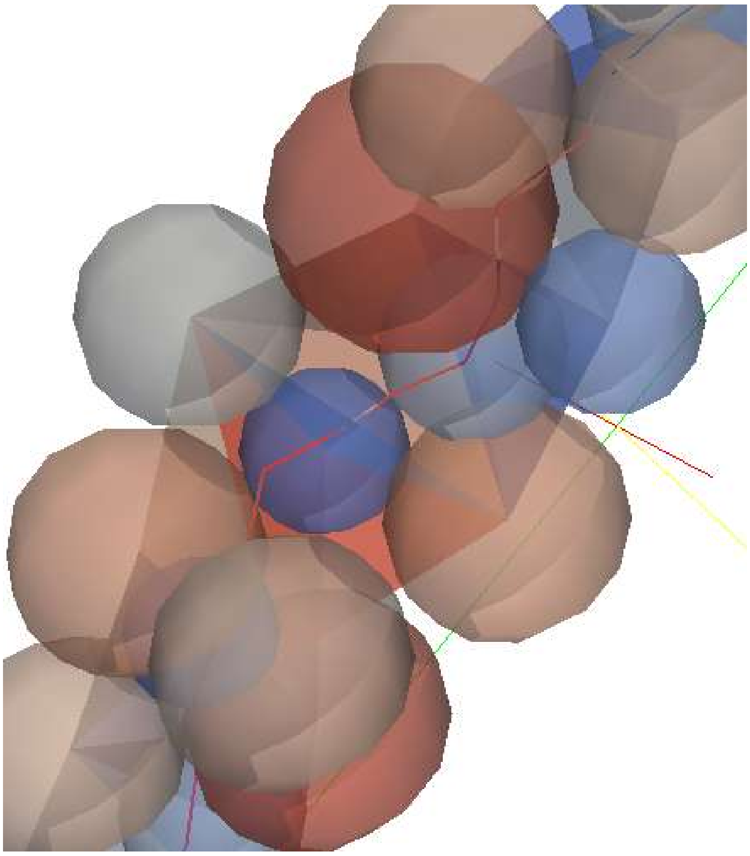
User's Guide

(for version IV.3)

by

Wojciech Sobieski, Seweryn Lipiński

University of Warmia and Mazury in Olsztyn
The Faculty of Technical Sciences



Olsztyn (Poland), May 2016

Table of Contents

1 The PathFinder project

1.1 Introduction

PathFinder – a program destined for analysis the spatial structure of granular beds, or its parts, of a cylindrical or cubical shape. The program is available free of charge with source code and can be used on Windows or Unix/Linux. The program needs additional software to work: gfortran compiler (for compiling source code), the Gnuplot environment (for basic visualization and analysis of the calculations) and ParaView (for the visualization and analysis of results). These programs are also available free of charge under the GPL license for both Windows and Unix/Linux.

Path Tracking Method (PTM) – iterative method of determination of the length of pore channel in the chosen space direction, which consists in analyzing the local structure of the pore space based on vector geometry.

In the case of Pathfinder code, the length of pore channel is determined between two parallel planes based on the sum of the unitary lengths, calculated based on so-called tetrahedral structures, which establish the basis for the calculation algorithm (Fig. 1.1). Tetrahedral structures are created based on the data on the location and diameter of each particle in the bed. The data for calculations are obtained from DEM simulation or from the analysis of a set of tomography scans. Details concerning method are available in the following subchapters.

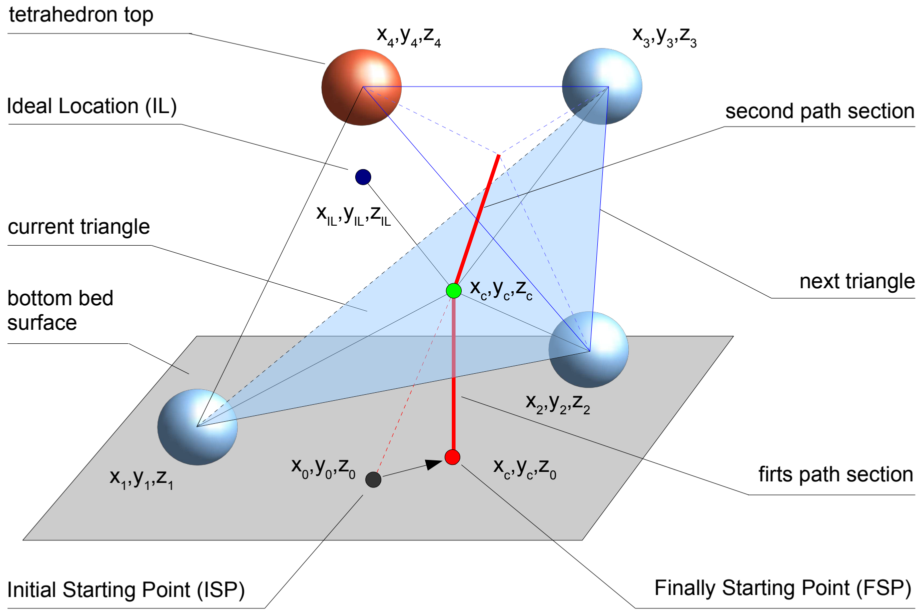


Fig. 1.1. Schema of the tetrahedral structure

Wojciech Sobieski developed PTM for granular beds in 2009 during an internship at the University of Manitoba in Winnipeg. The inspiration to start work on a method was observation, resulting from the author's extensive experience in programming, that DEM simulation results can be used to analyze the internal structure of porous media, in particular for the calculation of various geometrical parameters, including tortuosity. In 2009, the first working version of the program was created; the process of its creation was described in the Research Report No 1/2009 (in Polish) [18]. This report later became the basis for several articles in peer-reviewed journals. In the first phase of the project, the determination of tortuosity in porous media was investigated. Then other features were added. The PathFinder code is still under development and the team working on it grows every year.

Currently the PathFinder project is available on the Internet.

Home page of the project is www.uwm.edu.pl/pathfinder.

On the project page, interested Reader can find more information, as well as download the source code and additional tools.

1.2 The PathFinder limitations

Path Tracking Method was debveloped for the analysis of real beds, thus it has some limitations:

- spheres cannot be regularly arranged in a space (e.g. in the form of a structural mesh).
- variance of the spheres size distribution cannot be too high - tests have shown (using a bed of beads having an average diameter of 6 [mm]) that the program works properly for variations not higher than 1.1 [mm].

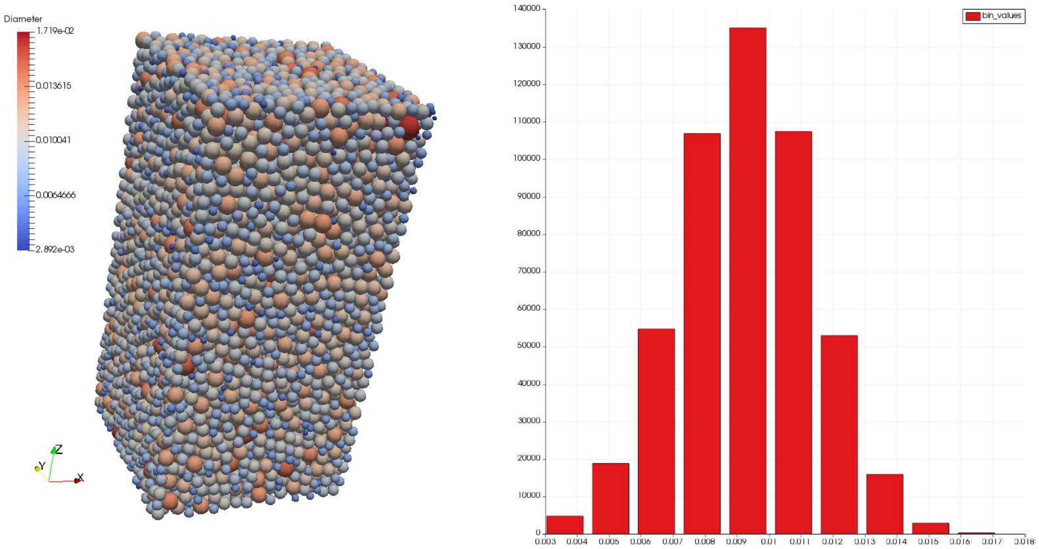


Fig. 1.2. An example of the deposit with a maximum possible variance

1.3 The PathFinder license

The conditions for using each of elements of the PathFinder project are as follows:

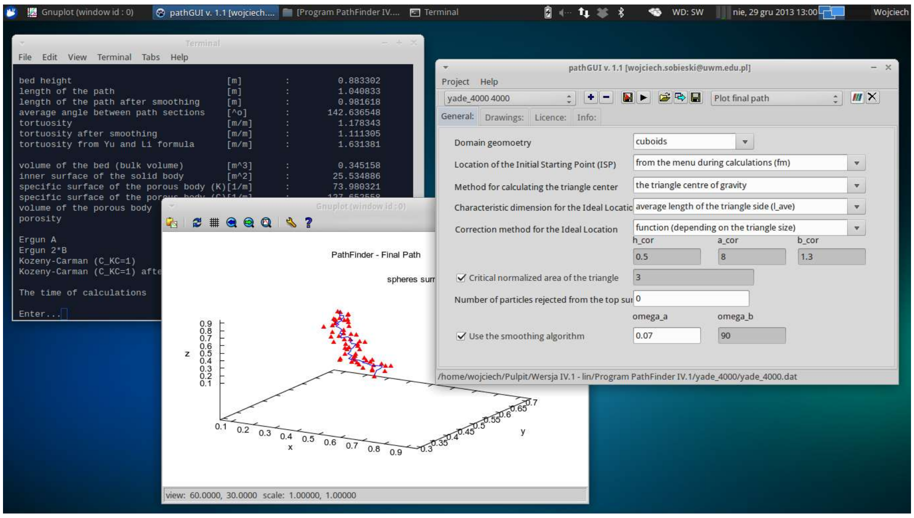
1. The software is free of charge and can be used for any purpose by institutions and individuals.
2. In case of publishing any material, for creation of which contributed the elements of the PathFinder project, in particular scientific articles, their authors are required to cite suitable articles from publications section, which is placed on the project website and in the User's Guide.
3. In case of redistribution of the software, no part of the installation package can be removed, in particular the terms of the license.
4. Copyright holders of each component of the project are indicated in the headers of the source code or in the documentation.
5. The authors of the project are not responsible for any damage or losses resulting from the use of software and the use of the information contained therein. The authors also assure that they have made every effort to make the project error-free and containing reliable information.

1.4 The PathFinder and Free Software

The project assumes that all its elements should be created using free tools and each of these tools should be available in both Windows (Fig. 1.3) and Unix/Linux (Fig. 1.4) if only it is possible.

Following applications are being used in the PathFinder project:

- gfortran – a Fortran language compiler;
- Edi – a self-made programming environment for gfortran compiler (in Polish, only for Windows);
- Gnuplot – graphical environment;
- ParaView – graphical post-processor;
- MayaVi – graphical post-processor;
- Lazarus – a RAD programming environment based on Free Pascal compiler;
- LibreOffice – an office suite.

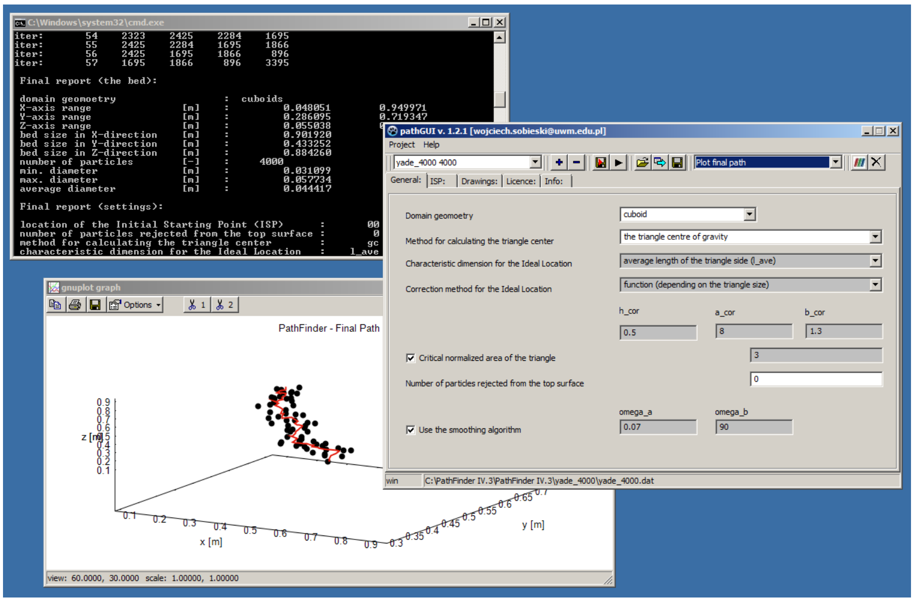

Gnuplot (window id : 0)
pathGUI v. 1.1 [wojciech....
[Program PathFinder IV....
Terminal
WD: SW
nie, 29 gru 2013 13:00
Wojciech
Terminal
File Edit View Terminal Tabs Help

```
bed height                              [m]      :        0.883302
length of the path                      [m]      :        1.040833
length of the path after smoothing      [m]      :        0.981618
average angle between path sections     [^o]     :      142.636548
tortuosity                              [m/m]    :        1.178343
tortuosity after smoothing              [m/m]    :        1.111305
tortuosity from Yu and Li formula       [m/m]    :        1.631381

volume of the bed (bulk volume)         [m^3]    :        0.345158
inner surface of the solid body         [m^2]    :       25.534886
specific surface of the porous body (K)[1/m]     :       73.980321
specific surface of the porous body (C)[1/m]     :      127.652558
volume of the porous body
porosity

Ergun A
Ergun 2*B
Kozeny-Carman (C_KC=1)
Kozeny-Carman (C_KC=1) afte

The time of calculations

Enter...
```

Domain geomoetry: cuboids
Location of the Initial Starting Point (ISP): from the menu during calculations (fm)
Method for calculating the triangle center: the triangle centre of gravity
Characteristic dimension for the Ideal Locatic: average length of the triangle side (l_ave)
Correction method for the Ideal Location: function (depending on the triangle size)
h_cor: 0.5 a_cor: 8 b_cor: 1.3
☑ Critical normalized area of the triangle: 3
Number of particles rejected from the top sur: 0
☑ Use the smoothing algorithm: omega_a: 0.07 omega_b: 90

/home/wojciech/Pulpit/Wersja IV.1 - lin/Program PathFinder IV.1/yade_4000/yade_4000.dat

Fig. 1.3. The PathFinder software running under Ubuntu Linux



```
iter:        54    2323    2425    2284    1695
iter:        55    2425    2284    1695    1866
iter:        56    2425    1695    1866     896
iter:        57    1695    1866     896    3395

 Final report (the bed):

 domain geomoetry                      :  cuboids
 X-axis range               [m]     :          0.048051       0.949971
 Y-axis range               [m]     :          0.286095       0.719347
 Z-axis range               [m]     :          0.055038       0
 bed size in X-direction    [m]     :          0.901920
 bed size in Y-direction    [m]     :          0.433252
 bed size in Z-direction    [m]     :          0.884260
 number of particles        [-]     :      4000
 min. diameter              [m]     :          0.031099
 max. diameter              [m]     :          0.057734
 average diameter           [m]     :          0.044417

 Final report (settings):

 location of the Initial Starting Point (ISP)      :      00
 number of particles rejected from the top surface :       0
 method for calculating the triangle center        :      gc
 characteristic dimension for the Ideal Location   :   l_ave
```

Domain geomoetry: cuboid
Method for calculating the triangle center: the triangle centre of gravity
Characteristic dimension for the Ideal Location: average length of the triangle side (l_ave)
Correction method for the Ideal Location: function (depending on the triangle size)
h_cor: 0.5 a_cor: 8 b_cor: 1.3
☑ Critical normalized area of the triangle: 3
Number of particles rejected from the top surface: 0
☑ Use the smoothing algorithm: omega_a: 0.07 omega_b: 90

win C:\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4000.dat

Fig. 1.4. The PathFinder software running under Windows 7

1.5 The PathFinder basics

The PathFinder software is intended for calculating a set of geometrical parameters of a porous bed, when the data with locations and sizes of all particles is available. PathFinder in fact is only a solver and so the visualization should be made in Gnuplot environment (during calculations) (Fig. 1.5) or in ParaView software (after calculations) (Fig. 1.6). As an alternative, the final visualization may be performed in MayaVi visualizer. Pathfinder automatically generates suitable output files for that purpose, in a form of simple text, VTK or CSV format, as well as shell scripts for operating system. Using other additional applications for results visualization is possible too.

The PathFinder code, as well as other described tools, may by used in Windows and UNIX/Linux. In both operating systems, the effects of the PathFinder code execution and the procedures themselves are the same.

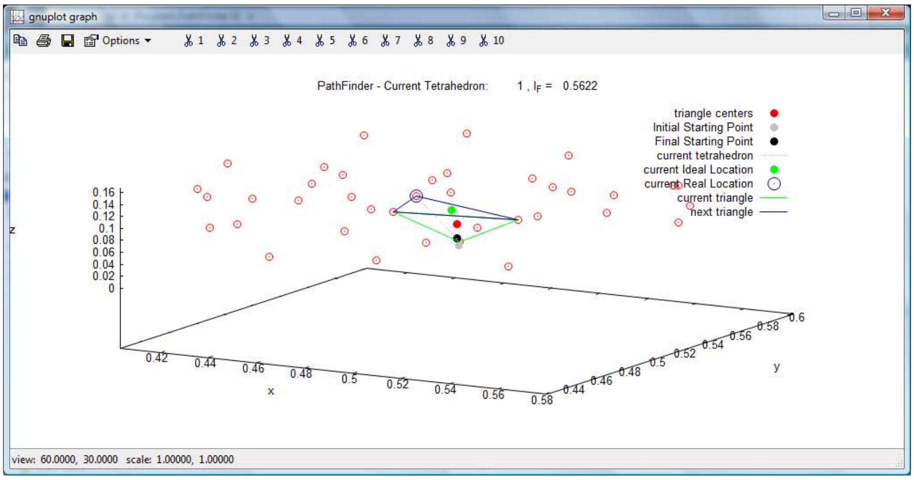


Fig. 1.5. Visualization of the first tetrahedron in a calculated path

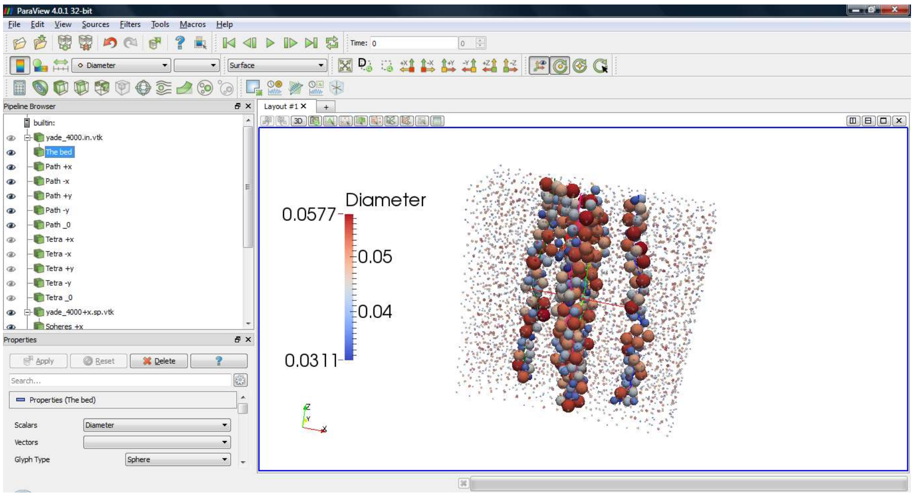


Fig. 1.6. Visualization of five paths calculated from different Initial Starting Points

1.6 The PathFinder package

The PathFinder is available for download from the project homepage as a compressed file. After acceptance of the license and package download, the file should be decompressed in a known directory, for which the user has full rights.

The general structure of the decompressed PathFinder folder (i.e. contained subfolders and files) is as follows:

- TheName – folder intended for the calculations, the project should be stored therein:
    - Comparison – subfolder in which the tortuosity distribution is collected, as well as files for additional calculation tools,
    - Comparison → isp_fc – subfolder in which data for deviation analysis when the ISP is defined by coordinates is collected,
    - Comparison → isp_fp – subfolder in which data for deviation analysis when the ISP is defined by the percents of the axis ranges is collected,
    - Comparison → isp_rg – subfolder in which data for deviation analysis for the so-called Regular Grid Method is collected,
    - Comparison → shift_XX_YY – subfolder in which data for deviation analysis for every pair of xshift and yshift parameters for the so-called Standard Method is collected,
    - isp_fc → subfolder with results of calculations for the current path,
    - isp_fp → subfolder with results of calculations for the current path,
    - isp_rg → subfolder with results of calculations for the current path,
    - shift_XX_YY – subfolder with results for every pair of xshift and yshift parameters for the so-called Standard Method,
    - TheName.dat – file with settings of the problem for PathFinder,
    - TheName.in – file with input data (see the description in Point 1.8),
- PathFinder.f90 – main source code of PathFinder for Windows and Unix/Linux;
- PathFinder.inc – additional Fortran code containing procedures needed to create scripts for data visualization in Gnuplot environment or the ParaView software;
- PathFinder.sys – settings file with the short name of the used OS: win or lin;
- PathFinder.exe – executable file for Windows OS;
- PathFinder.out – executable file for Unix/Linux OS;
- PathFinder.pdf – latest version of the User Guide;
- TheName.bat – script running the calculations in Windows OS;
- TheName.sh – script running the calculations in Unix /Linux OS.

Note, that there is no actual TheName folder and files after decompression. User should create it on his own, as well as its content, according to instructions contained in this User's Guide.

The package may also include: examples of calculation problems; graphical interface for PathFinder; additional calculation tools.

In the PathFinder software, a specification of the name of "TheName" folder and files is very important. It is used for creating a directory containing the problem for calculation, making scripts for running the calculations and as a call parameter.

TheName cannot contain spaces and should be short. It is advisable to include there a number of spheres used in problem being solved, for example: "pfc3d_181888" (181888 spheres) or "yade_4000" (4000 spheres).

After downloading and unpacking of the PathFinder package, the software may be used directly, without compiling of the source codes. However, there is one crucial issue – in Unix/Linux OS – all executable files (with *.out extension) and shell scripts (with *.sh extension) must have rights to execution in the system. It can be made using any graphical interface or with the following instruction in the console:

```
chmod +x TheName.sh
```

If something doesn't work correctly nonetheless, probably the compilation of source codes should help.

1.7 Compilation of the source code

The PathFinder code is written in Fortran 90/95 language and may be compiled by the gfortran compiler[1]. It is freely available for many operating systems and easily installable. For details read the gfortran documentation or the book [19] (available only in Polish).

The compilation of the source code:

- open the PathFinder.sys file in the main PathFinder directory;
- type “win” or “lin”, depending on operating system (OS) being used;
- open the shell console in the main PathFinder directory;
- in the console type

```
gfortran PathFinder.f90 -o PathFinder.exe
```

in Windows OS or

```
gfortran PathFinder.f90 -o PathFinder.out
```

in Unix/Linux OS.

The code was tested on Windows XP, Windows Vista, Windows 7 and Ubuntu Linux with Xfce graphical environment.

Make sure that the file can be executed in current location and with current user rights.

1.8 The PathFinder input file

The PathFinder software works with a text file with five columns containing information in following order: consecutive numbers of each sphere, X, Y and Z-coordinates of centres of spheres and diameters of spheres. The X and Y columns may be swapped, but the Z column must be always the fourth. This is important because the data after reading is sorted by the Z-coordinate. In data reading process, every line is considered as a new particle, so the number of lines gives information about the bed size. Below is an example of such file (TheName.in):

```
1      0.85692400      0.52585399      0.33372900      0.05773420
2      0.29979801      0.34492999      0.44405901      005772760
3      0.54170299      0.33786801      0.81188297      0.05772080
...
```

[1] GCC Home Page [on-line]. URL: http://gcc.gnu.org/fortran/

After running the program, input data is saved also in VTK and CSV file formats.

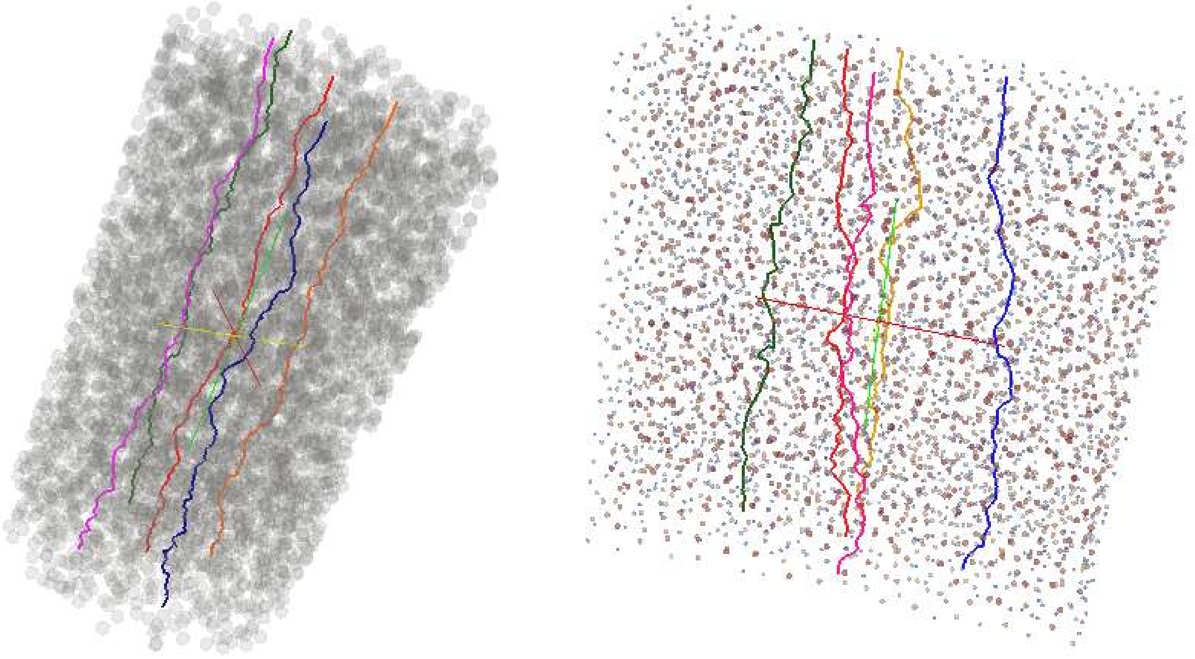


Fig. 1.7. Examples of visualized paths

For PathFinder it is not relevant in which way the input file was generated. Currently there are two main ways for obtaining appropriate data: using either results of the Discrete Element Method (DEM) simulation or results of Computed Tomography and 3D Image Analysis. Two examples from the first variant are shown in Fig. 1.7. The left presents a cylindrical domain with 18188 particles (DEM simulation was made in PFC^{3D} code[2]) and the right a cuboid domain with 4000 particles (DEM simulation was made in YADE code[3]). Points visible in both examples represent spheres of the bed, and the lines represent predicted paths of the fluid flow. The particles in both figures are rescaled (diameters are smaller) – so the paths are more visible.

Note that in cases when the source geometry is complex (in DEM models or of the objects scanned by using CT technique), only a representative part of the porous bed with appropriate shape can be exported for the PathFinder needs.

1.9 Additional software needed by PathFinder

The PathFinder needs Gnuplot[4] environment and ParaView[5] (or MayaVi Visualiser[6]) software for the visualization.

In case of using Gnuplot environment, it is important that Gnuplot scripts (with PLT extension) may be runned in the used operating system. The PathFinder automatically creates several scripts (into TheName directory) for every calculation problem, which may be used during calculations. The settings file TheName.dat contains parameters to configure the visualization process.

[2] ITASCATM Home Page [on-line]. URL: http://www.itascacg.com/pfc3d/

[3] YADE Home Page [on-line]. URL: https://yade-dem.org/wiki/Yade

[4] Gnuplot Home Page [on-line]. URL: http://www.gnuplot.info/

[5] ParaView Home Page [on-line]. URL: http://www.paraview.org/

[6] The MayaVi Data Visualizer [on-line]: URL: http://mayavi.sourceforge.net/

The Gnuplot environment and the ParaView software are both freely available in the Internet and very easy to install.

1.10 PathFinder versions

PathFinder IV.3 (2016-03-01):

- PathFinder – change in the definition of the so-called Quality Indicator;
- PathFinder – addition of a switch for saving the data for path deviation analysis;
- PathFinder – addition of two new Gnuplot scripts (Comp_np.plt and Comp_tau.plt) for comparing some results in the Standard Method;
- PathFinder – separation of Gnuplot scripts for visualisation of the tortuosity distribution (for the Standard Method) and the tortuosity field (for the Regular Grid Method);
- PathFinder – addition of a VTK file for visualisation of the tortuosity map;
- PathFinder – change in format of *.res files;
- PathFinder – change in format of the Final Report;
- PathFinder – correction of an error in the Final Report;
- PathFinder – correction of an error in the script "plot_current.plt";
- PathFinder – correction of an error in the Regular Grid Method for cases in with the Z axis is shifted;
- PathFinder – removal of the calculation of the tortuosity by Yu-Li formula;
- pathGUI – addition of different ISP schemes for cuboid and cylindrical beds as well as for the Regular Grid Method;
- Add-ons – addition of features of creating VTK files in the Convert-YADE tool;
- Add-ons – addition of a possibility to run the Convert-YADE tool with call paramethers;
- Add-ons – addition of a possibility to run Convert-YADE program by the use of call paramethers;
- Add-ons – addition of a new tool for calculation the characteristic dimension for the YADE code and the Particle Growing Method;
- Add-ons – addition of a new program for data conversion from the LIGGGHTS code to PathFinder.

PathFinder IV.2 (2015-11-09):

- PathFinder – addition of the so-called Regular Grid Method;
- PathFinder – addition of a new Gnuplot script for visualisation the tortuosity field;
- pathGUI – adaptation to the Regular Grid Method;
- User Guide – actualization of the documentation.

PathFinder IV.1.1 (2015-10-23):

- User's Guide – proposition of a general name of the approach developed by Wojciech Sobieski (Path Tracking Method);
- User's Guide – clarifying who is the Author of ideas described in the Research Report No 1/2009 [18];
- User's Guide – addition of the Chapter 5 (visualisation of results in the MayaVi software);
- User's Guide – updating the publication list.

PathFinder IV.1 (2014-05-07):

- PathFinder – addition of a new option for calculating the centre of the triangle, given as the geometric centre of the wet part of the triangle;

- PathFinder – addition of a new option, computing total area and wet area for each triangle;
- PathFinder – addition of a new option, computing total perimeter, wet perimeter and friction perimeter for each triangle;
- PathFinder – calculation of the specific surface area as defined by Kozeny and Carman
- PathFinder – addition of the possibility of calculating the deviation for path points
- PathFinder – addition of the option visualizing deviation of path points from the plumb-line
- PathFinder – addition of an option visualizing spatial step changes in path points on Z directions;
- PathFinder – addition of an option of adjusting the distance of ISP points from the centre of the basis of computational domain;
- PathFinder – addition of an automatic creation of subdirectories with the results for each pair of xshift and yshift (changes in the organization of files and directories);
- PathFinder – increased choice of the number of ISP points (from 5 to 9);
- PathFinder – change in organization of the selection of ISP points;
- PathFinder – addition of a option of calculating paths for all nine ISP points in one loop;
- Convert PFC^{3D} – changes in code organization;
- Convert PFC^{3D} – addition of a conversion option for file containing the data of diameters (previously only coordinates were converted);
- Sensitivity Calculator – addition of a choice of the definition of the specific surface area (as defined by Kozeny or Carman);
- User Guide – actualization of the documentation;
- User Guide – replacement of incorrect figures in the description of tortuosity calculation algorithm
- User Guide – addition of a description for the calculation of the specific surface area as defined by Kozeny and Carman

PathFinder IV.0 (2013-10-03):

- modification of the code and adapting it to work on Windows and Unix/Linux;
- organization of a code and changes in naming of variables and output files;
- reduction of the number of output files and data storage reorganization;
- update of PLT scripts and adapting them to the new version of Gnuplot environment;
- removal of the original graphical front end previously used for displaying results of calculations;
- addition of a calculation example, generated with the use of Discrete Element Method in YADE program;
- development (in Fortran language) of the program for converting YADE files to Pathfinder input format;
- development (in Fortran language) new version of the program for converting PFC3D files to Pathfinder input format;
- addition of a possibility of selection of computational domain type: cylindrical or cubical;
- addition of modules generating VTK and CSV files (for the needs of visualization in ParaView environment);
- development of ways to visualize the results in ParaView environment;
- addition of a User's Guide;
- development (in Lazarus environment) of program graphical interface (PathGUI program).

PathFinder III (2010-11-04):

- code modifications and adapting it to work with various computational problems;

- adaptation of the program to work with deposits of non-uniform particle diameters;
- organization and optimization of the source code;
- changes in organization of writing the results of calculations;
- changes in organization of visualization of results in Gnuplot environment.

PathFinder II (2010-02-05):

- development of algorithms to determine all the geometrical parameters present in Ergun and Kozeny-Carman laws;
- development of program for testing the sensitivity of Ergun and Kozeny-Carman models (Sensitivity Calculator);
- addition of a data storage module for the Sensitivity Calculator program;
- addition of a program for calculating the values of Ergun equation for air flow and parameters obtained from the PathFinder program (Ergun Calculator);
- addition of a program for calculating the values of Kozeny-Carman equation for air flow and parameters obtained from the PathFinder program (Kozeny-Carman Calculator);
- addition of a module for storing comparative data in a Comparison directory;
- removal of unnecessary fragments of the code.

PathFinder I (2009-12-10):

- development of the final version of the algorithm for calculating tortuosity in granular bed;
- development of a method for visualizing the results of calculations using scripts of Gnuplot environment;
- development of graphical front end (using Borland Delphi environment and OpenGL library) destined for displaying the results of calculations;
- development of the program (in Borland Delphi environment) for converting PFC3D files to Pathfinder input files.

2 The PathFinder algorithm

2.1 Calculating the domain range

All spheres in the bed are described by 5 variables (given as columns, as it was previously described):

- n_i – number of the ith sphere,
- x_i – X coordinate of the ith sphere centre,
- y_i – Y coordinate of the ith sphere centre,
- z_i – Z coordinate of the ith sphere centre,
- d_i – diameter of the ith sphere.

The total number of spheres in the bed is denoted by n_s .

In the first step, all spheres are sorted by the Z coordinate. In this way, the sphere lying lowest in the bed has the index $i=1$ and the highest sphere index is $i=n_s$. In determination of the Z-axis range, the particle diameters are taken into account (Fig. 2.1):

$$z_{min}=min\left(z_i-\frac{1}{2}\cdot d_i\right) \quad . \qquad (2.1)$$

Note, that if the bottom surface of the bed is horizontal and flat, many particles have the lowest surface point in the same XY plane.

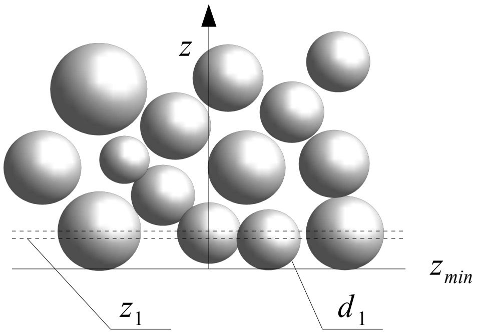


Fig. 2.1. Calculating the location of the bottom surface of the bed

In the same way the highest point in the bed on the top surface is calculated (Fig. 2.2)

$$z_{max}=max\left(z_i+\frac{1}{2}\cdot d_i\right) \qquad (2.2)$$

as well as ranges of X and Y axis

$$x_{min}=min\left(x_i-\frac{1}{2}\cdot d_i\right) \quad , \quad x_{max}=max\left(x_i+\frac{1}{2}\cdot d_i\right) \quad , \qquad (2.3)$$

$$y_{min}=min\left(y_i-\frac{1}{2}\cdot d_i\right) \quad , \quad y_{max}=max\left(y_i+\frac{1}{2}\cdot d_i\right) \quad . \qquad (2.4)$$

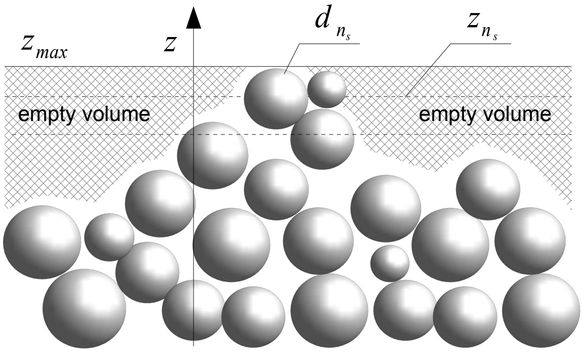


Fig. 2.2. Calculating the location of the top surface of the bed

In real beds, the bottom and the lateral surfaces usually limit the locations of particles. However, the top surface can be uneven and some particles may protrude from the bed. In this case, "empty area" in the bed exists, what may give inaccuracies in calculations. To avoid this, all protruding particles should be discarded from the calculations (Fig. 2.3). The number of discarded particles is described by the variable n_{s_rej} . Also, notice the small red particle right to the particle determining the height of the bed (before discarding protruding particles). Despite it has the highest value of Z coordinate, due to its small diameter it is not recognized by the algorithm as the highest sphere in the bed.

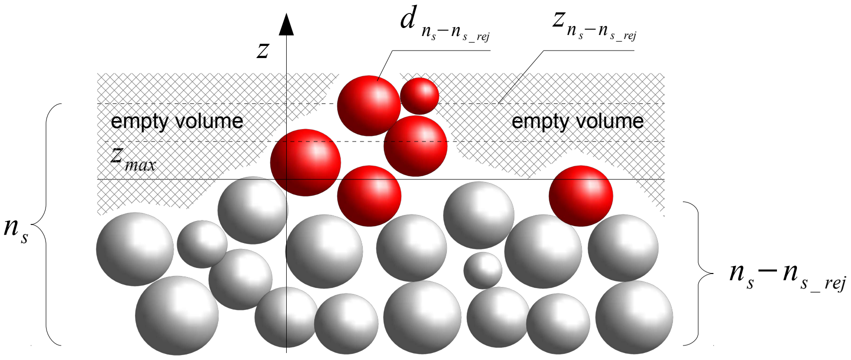


Fig. 2.3. Visualization of the idea of rejecting protrudes particles: the empty volume is not taken into account

2.2 Calculating the default Initial Starting Point

The default Initial Starting Point, from which the algorithm starts, is calculated as

$$x_0=x_{min}+\frac{x_{max}-x_{min}}{2} \quad , \quad y_0=y_{min}+\frac{y_{max}-y_{min}}{2} \quad , \quad z_0=z_{min} \quad , \tag{2.5}$$

where index 0 denotes the ISP.

In the PathFinder source code, the same variables are then used for the Final Starting Point.

2.3 Calculating the domain height

The domain height is calculated as

$$L_0=z_{n_s-n_{s\_rej}}+0.5\cdot d_{n_s-n_{s\_rej}}-z_{min} \quad , \tag{2.6}$$

where:

L_0 - the domain height [m],
$z_{n_s-n_{s\_rej}}$ - the Z coordinate of the highest used in calculations sphere in the bed [m],
$0.5\cdot d_{n_s-n_{s\_rej}}$ - radius of the highest used in calculations sphere in the bed [m],
z_{min} - the Z coordinate of the bottom surface of the bed [m].

2.4 Calculating the domain volume

In the PathFinder code calculations can be performed in porous beds of cuboid or cylindrical shape. In the first case the volume of the bed is calculated from the formula:

$$V=(x_{max}-x_{min})\cdot(y_{max}-y_{min})\cdot\left[z_{n_s-n_{s\_rej}}+0.5\cdot d_{n_s-n_{s\_rej}}-z_{min}\right] \quad , \tag{2.7}$$

where:

V - volume of the porous bed (the domain) [m^3],
x_{max} - the furthest point of the sphere surface in the positive direction of the X axis [m],
x_{min} - the furthest point of the sphere surface in the negative direction of the X axis [m],
y_{max} - the furthest point of the sphere surface in the positive direction of the Y axis [m],
y_{min} - the furthest point of the sphere surface in the negative direction of the Y axis [m],
$z_{n_s-n_{s\_rej}}$ - the Z coordinate of the highest sphere used in calculations [m],
$d_{n_s-n_{s\_rej}}$ - the diameter of the highest sphere used in calculations [m],
z_{min} - the Z coordinate of the bottom surface of the bed [m],
n_s - total number of spheres (equal to the number of lines in the input data file) [-],
n_{s_rej} - a number of spheres discarded from the calculations [-].

When the porous bed has a cylindrical shape, the domain volume is calculated as follows:

$$V=\frac{\pi\cdot d_{cyl}^2}{4}\cdot\left[z_{n_s-n_{s\_rej}}+0.5\cdot d_{n_s-n_{s\_rej}}-z_{min}\right] \quad , \tag{2.8}$$

where d_{cyl} is the diameter of the cylinder [m], calculated as the average range of X and Y axis:

$$d_{cyl}=\frac{1}{2}\cdot\left[(x_{max}-x_{min})+(y_{max}-y_{min})\right] \quad . \tag{2.9}$$

Note, that the expression in square brackets in formulas (2.7) and (2.8) is equal to the height of the domain.

2.5 Calculating the tortuosity

Tortuosity is the ratio of the actual length of flow path to the physical depth of a porous bed:

$$\tau=\frac{L_p}{L_0} \quad , \tag{2.10}$$

where:

L_p - the length of the flow path [m],
L_0 - the depth of the porous bed [m].

The algorithm for calculating the length L_p in PathFinder was described in details in the report [18] and next in the paper [9]. Summarized procedure is presented below:

a) arbitrarily choose an Initial Starting Point (ISP) at the bottom of the porous bed (Fig. 2.4);
b) find three nearest particles to the ISP (Fig. 2.5) that form a triangle in the space (Fig. 2.6);
c) calculate the coordinates of the centre of gravity of the surface (in the triangle plane) through which fluid flows (Fig. 2.7);
d) move the Initial Starting Point to the Final Starting Point (FSP) (Fig. 2.8) - in this way the first path section is perpendicular to the bottom surface of the bed;
e) calculate the normal to the triangle, in the direction of Z axis (Fig. 2.9);
f) estimate coordinates of so called Ideal Location (IL), in which should be located the next sphere surrounding the path (Fig. 2.10);
g) move the Ideal Location closer to the triangle centre (Fig. 2.11) (details in Point 2.7);
h) find the nearest particle to the Ideal Location (2.12) - this is the Real Location (RL) of the 4-th particle forming tetrahedron in the space (Fig. 2.13);
i) remove the lowest sphere from tetrahedron 1-2-3-4 to obtain the base triangle for the next tetrahedron (Fig. 2.14);
j) continue the calculations from the point c), until reaching the top surface of the bed;
k) calculate the length of the path inside the tetrahedron (a flow path is a line connecting the centroid of the base triangle with the centroid of one of the three side triangles) (Fig. 2.15);
l) smooth the path and calculate the path length L_p .

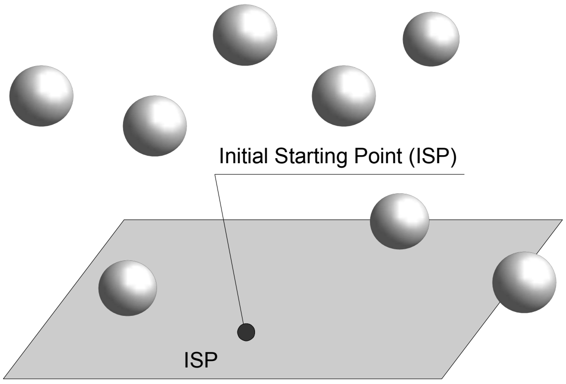


Fig. 2.4. An Initial Starting Point

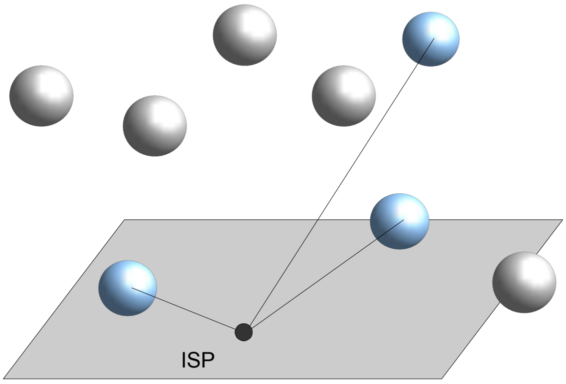


Fig. 2.5. Three nearest particles to the ISP

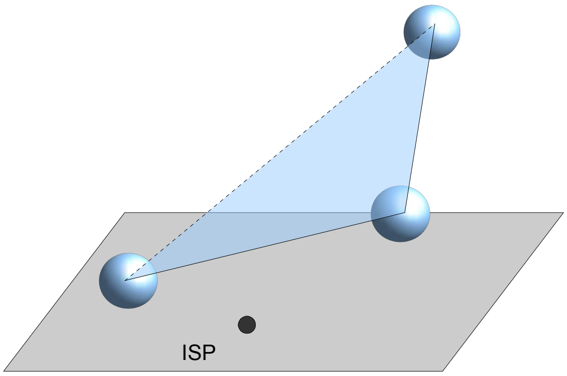


Fig. 2.6. Triangle formed by the particles

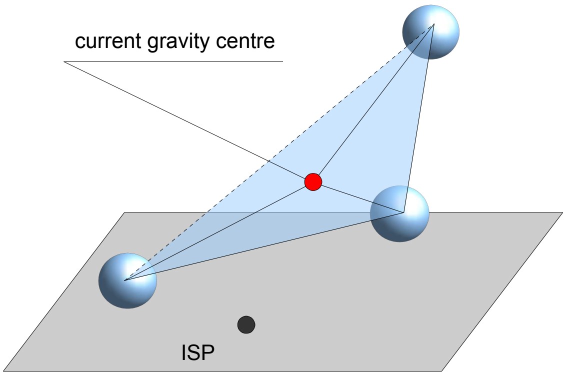


Fig. 2.7. The gravity centre of the current triangle

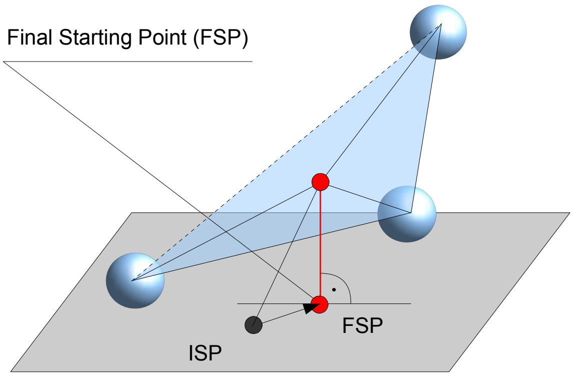


Fig. 2.8. The Final Starting Point

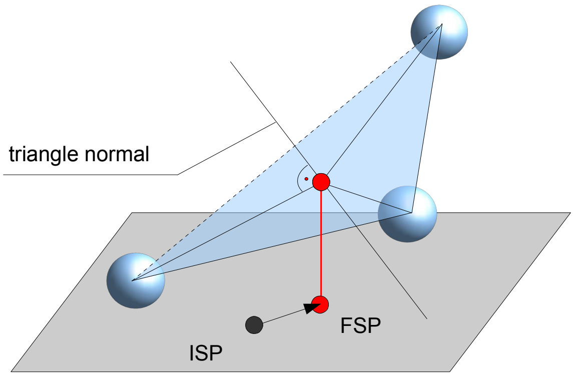


Fig. 2.9. The triangle normal

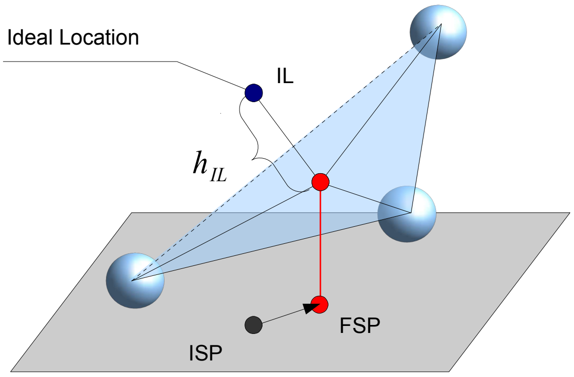


Fig. 2.10. The Ideal Location

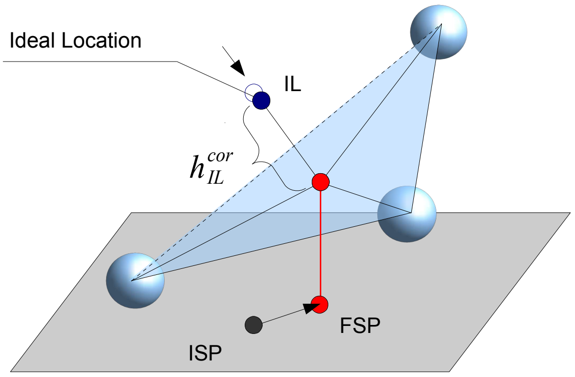


Fig. 2.11. The Ideal Location after correction

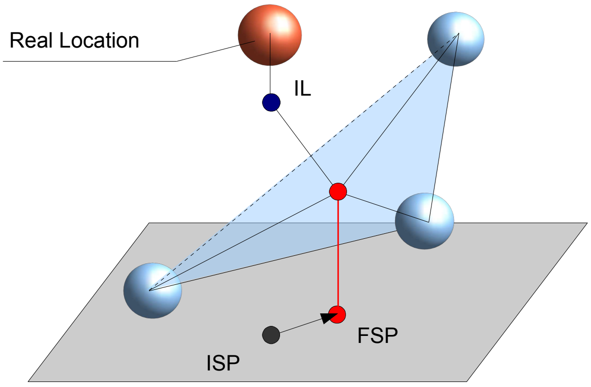


Fig. 2.12. The sphere nearest to the IL

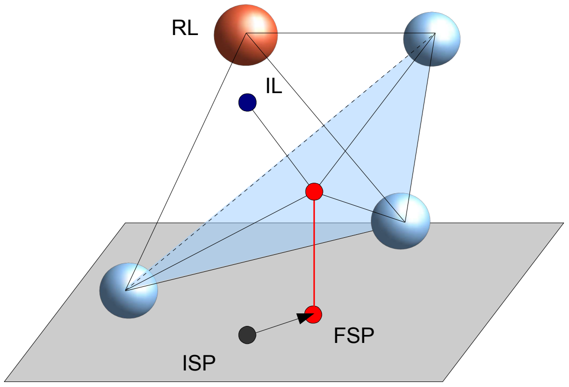


Fig. 2.13. The tetrahedron

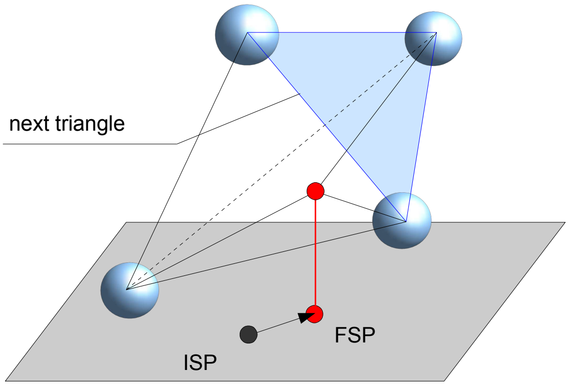


Fig. 2.14. The next triangle

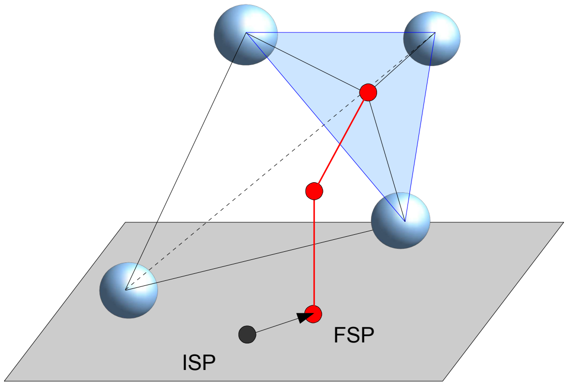


Fig. 2.15. Path sections

The algorithm stops when the distance between the gravity centre and the Ideal Location is smaller than the distance between the top surface of the bed and the Z coordinate of the Ideal Location.

$$l_{min} < z_{max} - z_{il} \quad , \tag{2.11}$$

The last section is perpendicular to the top surface of the bed.

The number of path points is bigger by two than the number of tetrahedrons used for searching the path sections.

2.6 The coordinates of the centroid of the triangle

The algorithm for calculating the centre of gravity of the shape, through which fluid flows is described in details in the work [4]. This algorithm contains of two steps. At first, coordinates of all basic figures are calculated (one triangle and three circle segments) and next the summation method is used to calculate coordinates of the resultant centre of gravity of the set of basic shapes (Fig. 2.16)

$$\begin{aligned} x_{gc} &= \frac{A_t \cdot x_{tc} - A_1 \cdot x_{S1} - A_2 \cdot x_{S2} - A_3 \cdot x_{S3}}{A_t - A_1 - A_2 - A_3} \\ y_{gc} &= \frac{A_t \cdot y_{tc} - A_1 \cdot y_{S1} - A_2 \cdot y_{S2} - A_3 \cdot y_{S3}}{A_t - A_1 - A_2 - A_3} \\ z_{gc} &= \frac{A_t \cdot z_{tc} - A_1 \cdot z_{S1} - A_2 \cdot z_{S2} - A_3 \cdot z_{S3}}{A_t - A_1 - A_2 - A_3} \end{aligned} \tag{2.12}$$

where:

x_{gc} , y_{gc} , z_{gc} - coordinates of the obtained centre of gravity [m],
A_t - area of the current triangle [m^2],
x_{tc} , y_{tc} , z_{tc} - coordinates of the current triangle centre [m],
$A_{1,2,3}$ - areas of circle segments [m^2],
x_s , y_s , z_s - coordinates of gravity centres (denoted by the S sign) of circle segments [m].

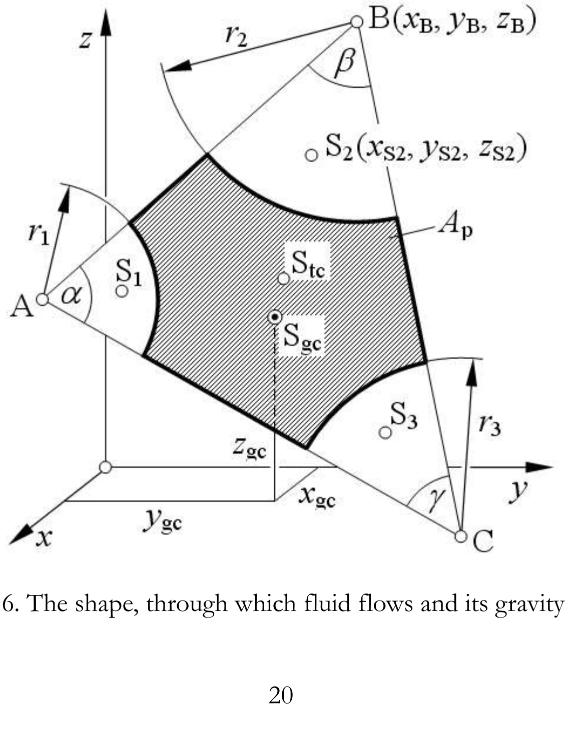


Fig. 2.16. The shape, through which fluid flows and its gravity centre

2.7 The Ideal Location and its correction

For calculating the Ideal Location, a characteristic dimension is used. It can be the average length of the current triangle sides l_{ave} (recommended) or the average diameter of spheres d_{ave} forming the current triangle. In consecutive calculations it was assumed, that the tetrahedron is regular and all sides have lengths equal to the characteristic dimension. In that case, the height of such tetrahedron may be calculated from the simple formula

$$h_{IL}=c_{dim}\cdot\sqrt{\frac{2}{3}} \quad , \tag{2.13}$$

where:

h_{IL} - the predicted distance between the IL and the triangle plane [m],
c_{dim} - the characteristic dimension [m].

The height h_{IL} is measured on the triangle normal in the direction of increasing values of the Z coordinates.

In real systems, particles forming a triangle may do not touch each other, what causes that the fourth particle falls partly between them. In extreme cases, the fourth particle can slide completely between particles forming the triangle. That can happens if the actual triangle area is three times larger than the area of a triangle with side equal to the average diameter of the spheres forming the current triangle. In that case a correction algorithm is used (it may be turned off using the settings file) and three new particles close to the current triangle centre are then searched. The number of such corrected triangles is shown in the final report. This normalized critical area of the triangle is also included in the settings file, but it should not be changed.

Usually the area of the real triangle is far from the critical area and the "sinking" effect is small, but not to omit. Therefore, the formula (2.13) should rather be defined as follows

$$h_{IL}=c_{dim}\cdot h_{cor}\cdot\sqrt{\frac{2}{3}} \quad , \tag{2.14}$$

where h_{cor} is a correction coefficient for Ideal Location [-]. This coefficient may be defined (in the settings file) as a constant value

$$h_{cor}=0.5 \quad , \tag{2.15}$$

or a function

$$h_{cor}=f\left(I_A\right)=1-\frac{\exp\left(a\cdot\left(I_A-b\right)\right)}{1+\exp\left(a\cdot\left(I_A-b\right)\right)} \quad , \tag{2.16}$$

where:

I_A - the area indicator [-],
a - a coefficient (obtained in earlier numerical investigations [18]), respondent for the function gradient and equal to 8 [-], b - a coefficient (obtained in earlier numerical investigations [18]), equal

to average value of normalized triangle area for the whole porous bed, with value equal to 0.5 [-].

The area indicator is defined as

$$I_A = \frac{A_i}{A_0} \quad , \tag{2.17}$$

where the current triangle area A_i is calculated from the Heron formula

$$A_i = \sqrt{\frac{L}{2} \cdot \left(\frac{L}{2} - a\right) \cdot \left(\frac{L}{2} - b\right) \cdot \left(\frac{L}{2} - c\right)} \tag{2.18}$$

and the area of reference triangle A_0 (triangle with all sides equal to the characteristic dimension) as

$$A_0 = \frac{c_{dim} \cdot \sqrt{3}}{4} \quad . \tag{2.19}$$

Symbols a , b and c in formula (2.18) denotes lengths of current triangle sides [m].

The details about this issue as well as the way of obtaining all constants are described in a research report [18], which was written parallel to performing implementation actions. This report contains descriptions of many issues not mentioned here and shows the process of creation the final PathFinder code.

2.8 Calculating the porosity

Having information about diameters of all particles in the bed, the volume of the solid body can be calculated as follows

$$V_s = \sum_{i=1}^{n_s - n_{s\_rej}} \frac{1}{6} \cdot \pi \cdot d_i^3 \quad , \tag{2.20}$$

where:

V_s - volume of the solid part of the porous body [m^3],
d_i - the diameter of the ith particle in the bed [m].

Using the formula (2.7) or (2.8) and the equation (2.20), the porosity of the granular bed may be consequently calculated as

$$e = \frac{V_p}{V} = 1 - \frac{V_s}{V} \quad , \tag{2.21}$$

where:

e - porosity (the ratio of the pore volume to the volume of the domain) [-], V_p - volume of the pore part of a porous body [m^3].

2.9 Calculating the inner and the specific surface of the solid body

The inner surface of the solid body may be obtained from the formula

$$S_s = S_p = \sum_{i=1}^{n_s - n_{s\_rej}} \pi \cdot d_i^2 \quad , \tag{2.22}$$

where:

S_s - the inner surface of the solid body [m^2], S_p - the surface of the solid body [m^2].

Equation (1.17) is true under the assumption that the particles forming the bed are rigid – in that case, contact between them is limited to the points. In the case of deformable particles, formula (1.17) describes the theoretical maximum surface area (actual inner surface of such a deposit is smaller, due to the presence of non-zero contact surfaces).

The specific surface of the porous body is calculated in two ways, according to the Kozeny theory [1,2]

$$S_{0,Kozeny} = \frac{S_s}{V} \tag{2.23}$$

and to the Carman theory [5]

$$S_{0,Carman} = \frac{S_s}{V_s} \quad , \tag{2.24}$$

where:

S_0 - the specific surface of the porous body [1/m].

The relationship between both definition is as follows

$$\frac{S_{0,Kozeny}}{S_{0,Carman}} = \frac{\frac{S_s}{V}}{\frac{S_s}{V_s}} = \frac{V_s}{V} = \frac{V \cdot (1-e)}{V} = 1 - e \quad . \tag{2.25}$$

The approach to the definition of the specific surface of the porous body is one of the differences between the Kozeny equation and the Carman formula [5].

2.10 Smoothing the path

The PathFinder allows using an optional smoothing algorithm, which shortens the length of the path. It was introduced on the basis of a fact, that fluid cannot turn suddenly at a point (Fig. 2.17).

The length of the smoothed path is calculated as follows (Fig. 2.18):

$$L_p^{smooth} = \frac{1}{2} \cdot \left(a_1 + b_{n_p}\right) + \sum_{2}^{n_p - 1} \frac{1}{2} \left(a_i + b_i\right) \cdot f\left(\omega_i\right) \quad , \tag{2.26}$$

where:

L_p^{smooth} - the length of the smoothed path [m],
a_1 - the length of the first path section [m],
b_{n_p} - the length of the last path section [m],
a_i - the length of first path section forming the ith angle [m],
b_i - the length of second path section forming the ith angle [m],
ω_i - the angle between two path sections [°],
n_p - the total number of path sections [-],
$f(\omega_i)$ - a function of ω_i [-].

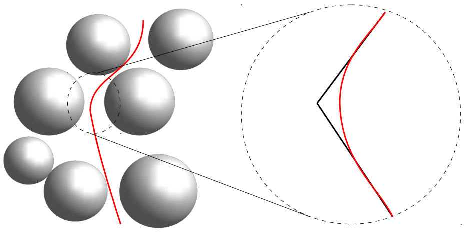


Fig. 2.17. The idea of path smoothing

The degree of shortening the length of the path depends on the current angle and it grows as more acute is following angle

$$f(\omega_i)=\frac{\exp\left(\omega_a\cdot(\omega_i-\omega_b)\right)}{1+\exp\left(\omega_a\cdot(\omega_i-\omega_b)\right)} \quad , \qquad (2.27)$$

where:

ω_a - model constant [°],
ω_b - constant equal to the right angle (90) [°].

For $\omega_i=180$ [°] the value of $f(\omega_i)$ is equal to one.

More details of the theoretical basis of the PathFinder code the Reader can find in work [9] as well as in the research report [18].

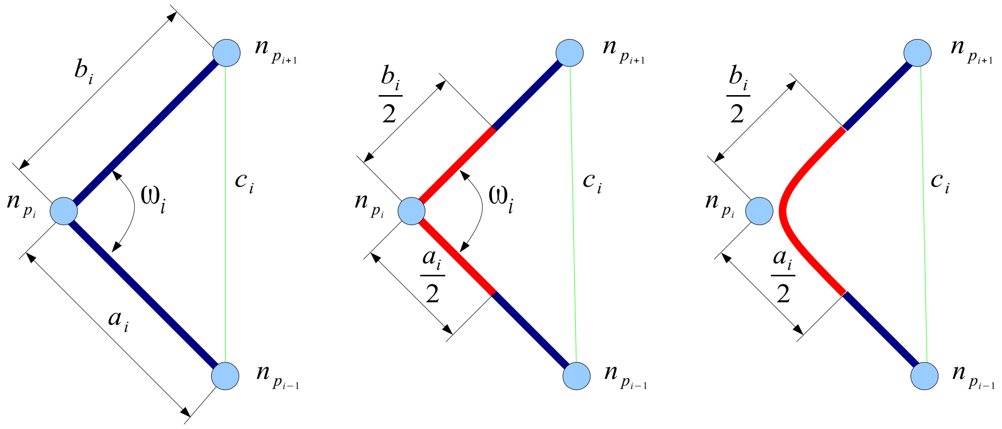


Fig. 2.18. The smoothing method

3 New calculation project

3.1 Basic steps

To start solving a new problem, following steps should be undertaken:

- create a directory called TheName in the main PathFinder directory;
- create the input file (TheName.in) inside TheName directory (see the Point 1.8);
- create the settings file (TheName.dat) inside TheName directory;
- optionally create a script (TheName.bat or TheName.sh) running the calculations, in the main PathFinder directory.

3.2 Calculations settings

All settings are collected in the TheName.dat file located inside TheName directory. It is a simple text file and can be easily edited using for example Notepad in Windows or Kate in Unix/Linux.

The settings file TheName.dat may be copied from an example included in the PathFinder package, created manually or created using a graphical interface (see point 4.1).

Typical content of the settings file is as follows:

```
Y       !lic - acceptance of the licence terms: Y (yes), N (no)
cub     !domain - domain geometry: cub (cuboids), cyl (cylinder)
al      !isp ISP location: 00, +-, 0-, --, -0, -+, 0+, ++, ff (file), fm (menu), al (SM), rg (RGM)
0.5     !xshift - shift of the ISP from the axis centre in X direction
0.5     !yshift - shift of the ISP from the axis centre in Y direction
0       !ns_rej - number of particles rejected from the top surface
2       !tr_cen - method for calculating the triangle center
1       !char_dim - characteristic dimension for the IL: 1 (l_ave), 2 (d_ave), 0 (ask for method)
2       !il_cor - correction method for the IL: 1 (constant value), 2 (function), 0 (ask for method)
0.5     !il_c_cor - correction coefficient for the constant method
8       !il_a_cor - function parameter a for the function method
1.3     !il_b_cor - function parameter b for the function method
Y       !cr_ar_cor - correction based on critical area of the triangle: Y (yes), N (no)
3       !cr_ar - critical area of the triangle
Y       !smooth - smoothing: Y (yes), N (no)
0.07    !omega_a - a coefficient in the smoothing algorithm
90      !omega_b - b coefficient in the smoothing algorithm
N       !draw_d - drawing particle diameters: Y (yes), N (no)
N       !draw_first - drawing the first tetrahedron:
N       !draw_current - drawing tetrahedrons during calculations or not: Y (yes), N (no)
N       !draw_area - drawing the areas of triangles: Y (yes), N (no)
N       !draw_perimeter - drawing the perimeters of triangles: Y (yes), N (no)
N       !draw_ch - drawing values of correction coefficient (for C_h or C_h_f): Y (yes), N (no)
N       !draw_if - drawing the quality indicator IF: Y (yes), N (no)
N       !draw_normal - drawing angle between normal and Z-axis: Y (yes), N (no)
Y       !draw_final - drawing the final path: Y (yes), N (no)
N       !save_csv - save the CSV files: Y (yes), N (no)
Y       !save_dev - save files with path points deviation: Y (yes), N (no)
End
```

NOTE: do not change parameters values marked with the grey background.

It is very important to set the shape of the domain. If you make a mistake, the porosity will be calculated erroneously. Details considering all the parameters one can find in article [9].

3.3 Running the calculations

To run a new calculations go to the main PathFinder directory and in the console, type:

```
PathFinder.exe TheName number_of_spheres
```

in Windows or

```
PathFinder.out TheName number_of_spheres
```

in Unix/Linux.

The PathFinder software needs to work with two call parameters: TheName and the number of spheres saved in TheName.in file.

To simplify calling the program appropriate shell scripts may be created: BAT files in Windows and SH files in Unix/Linux.

An example of the BAT file in Windows:

```
PathFinder.exe yade_4000 4000
pause
```

An example of the SH file in Unix/Linux:

```
xfce4-terminal -x './PathFinder.out yade_4000 4000'
```

The name of terminal in Unix/Linux operating system depends on the used graphical environment. The above example was used in Ubuntu Linux with the Xfce graphical environment.

The Initial Starting Point (ISP) must be chosen for performing the calculations. Currently such options are available:

- **Choosing ISP directly in the settings file as one of 9 default locations.** The description of point location contains two signs (3.1): the first refers to the X axis, the second to the Y axis. Both signs may have values: "–" (the negative part of the axis), "0" (zero point of the axis) and "+" (the positive part of the axis). The combination of signs is as follows: +0, +–, 0–, ––, –0, –+, 0+, ++, 00.

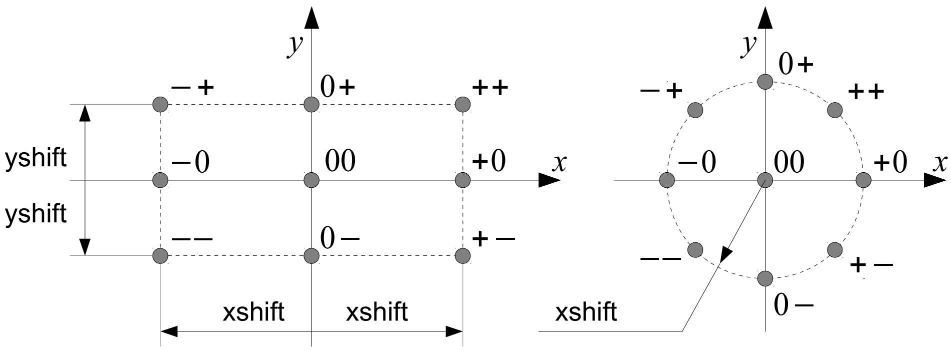
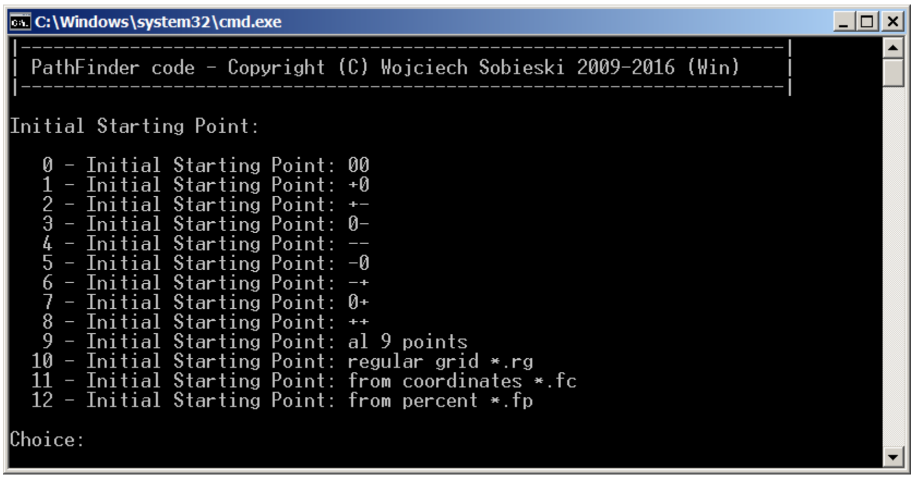


Fig. 3.1. Default locations of the Initial Starting Point for both types of domain

By default, all points surrounding the zero point of X and Y axis are located in the middle of the segment connecting the zero point and the domain wall. This distance may be changed using xshift and yshift options in the settings file. These coefficients may be between 0 (the ISP located in the zero point – the middle of the bottom domain wall) and 1 (the ISP located on the wall). The default values for both coefficients are equal to 0.5. In cylindrical domain only the xshift is used – to define the distance between the zero point and the wall.

- **Choosing ISP after running the program as one of 9 default locations** (Fig. 3.2). To use this option, the ISP method must be specified as “fm” (from menu) in the settings file.



```
|-----------------------------------------------------------------|
| PathFinder code - Copyright (C) Wojciech Sobieski 2009-2016 (Win) |
|-----------------------------------------------------------------|

Initial Starting Point:

   0 - Initial Starting Point: 00
   1 - Initial Starting Point: +0
   2 - Initial Starting Point: +-
   3 - Initial Starting Point: 0-
   4 - Initial Starting Point: --
   5 - Initial Starting Point: -0
   6 - Initial Starting Point: -+
   7 - Initial Starting Point: 0+
   8 - Initial Starting Point: ++
   9 - Initial Starting Point: al 9 points
  10 - Initial Starting Point: regular grid *.rg
  11 - Initial Starting Point: from coordinates *.fc
  12 - Initial Starting Point: from percent *.fp

Choice:
```

- Fig. 3.2. Choice of Initial Starting Points in the menu

- **Running calculations for all characteristic Initial Starting Points in one loop**. In this case, the ISP method must be specified as “al” (all). Then all 9 paths (+0, +–, 0–, —, –0, –+, 0+, ++, 00) will be calculated and saved in a subdirectory in TheName folder. The use of this approach three times, with xshift and yshift equal to 0.25, 0.5 and 0.75, is here called the Standard Method.

- **Using the so-called Regular Grid Method**. In this approach, both ranges of coordinates in the X and Y directions are divided into n_x and n_y points, respectively. Next, in a loop, all these points are used as ISPs. With this method, for one bed, $n_x \times n_y$ paths lengths may be calculated.

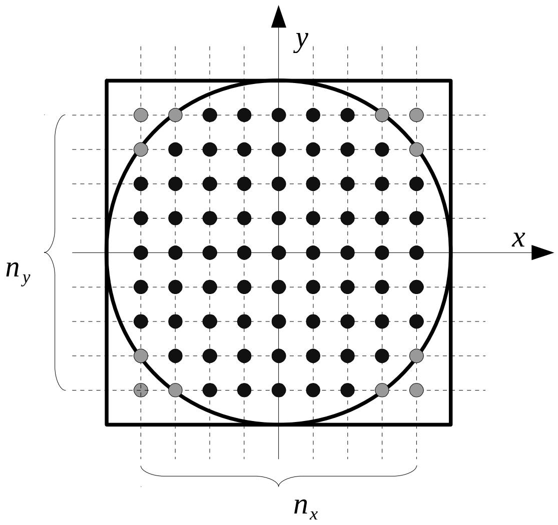


Fig. 3.3. Defining the Initial Starting Point in the Regular Grid Method

In this case, a file TheName.rg in TheName directory must exist. This is a text file with two integer values equal to n_x and n_y . The coordinates of next ISPs are calculated based on the ranges of X and Y axes and the appropriate steps in space. The minimum value on the Z-axis in whole bed is used as the Z coordinate. For example:

```
10 10
```

To use this option, the ISP method must be specified as “rg” in the settings file.

If option “rg” for location of the ISP is set in the TheName.dat file, the question about choice of the ISP location will not be displayed.

- **Defining ISP in a file using the “fc” (from coordinates) setting** (Fig. 3.4). In this case a file TheName.fc in TheName directory must exist. This is a text file with two real or double values equal to the X and Y-coordinates of the initial point (the minimum value on the Z-axis in whole bed is used as the Z coordinate). For example:

```
0.1 0.04
```

If option “fc” for location of the ISP is set in the TheName.dat file, the question about choosing the ISP location will not be displayed.

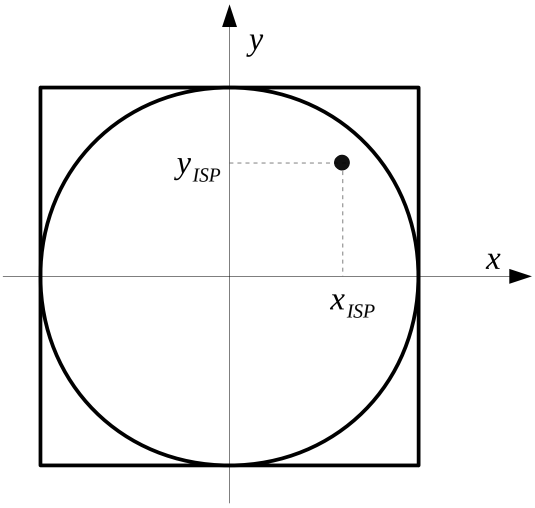


Fig. 3.4. Defining the Initial Starting Point by coordinates

- **Defining ISP in a file using the “fp” (from percent) setting** (Fig. 3.5). In this case a file TheName.fp in TheName directory must exist. This is a text file with two real/double values, equal to the percents of ranges of the coordinates in X and Y direction (the minimum value on the Z-axis in whole bed is used as the Z coordinate). For example:

```
25 25
```

If option “fp” for location of the ISP is set in the TheName.dat file, the question about choosing the ISP location will not be displayed.

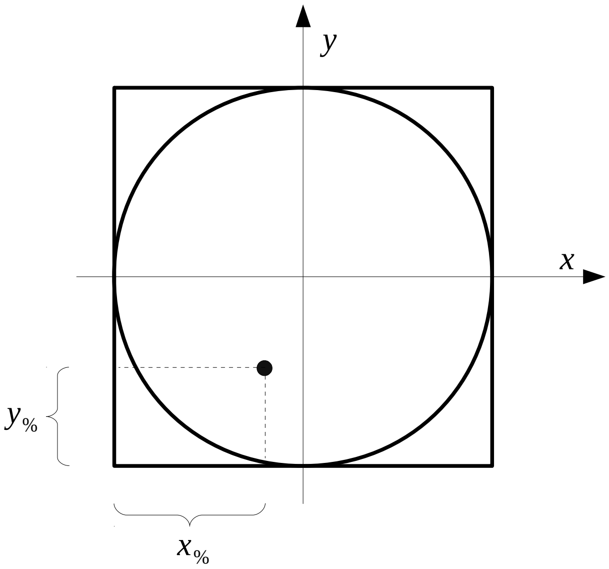
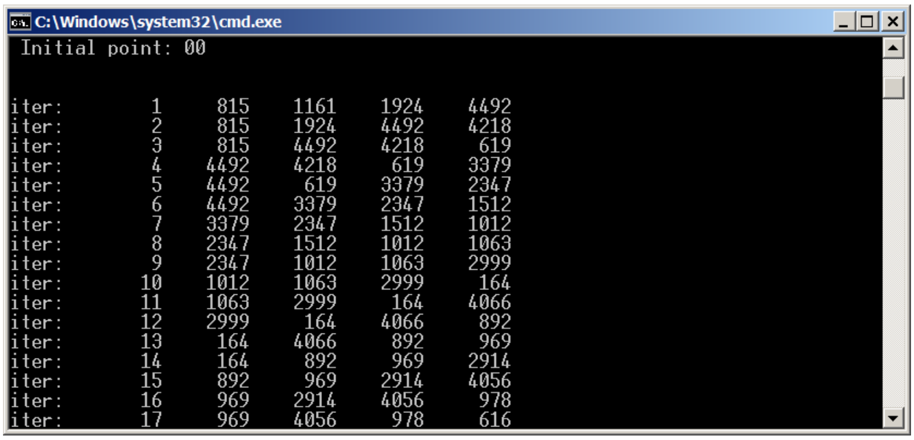


Fig. 3.5. Defining the Initial Starting Point by percents

After defining the method of ISP choice, the iteration process is initialized. During calculations, number of the current tetrahedron and the numbers of spheres forming the current tetrahedron are printed in the console (Fig. 3.6).



```
 Initial point: 00


iter:         1     815    1161    1924    4492
iter:         2     815    1924    4492    4218
iter:         3     815    4492    4218     619
iter:         4    4492    4218     619    3379
iter:         5    4492     619    3379    2347
iter:         6    4492    3379    2347    1512
iter:         7    3379    2347    1512    1012
iter:         8    2347    1512    1012    1063
iter:         9    2347    1012    1063    2999
iter:        10    1012    1063    2999     164
iter:        11    1063    2999     164    4066
iter:        12    2999     164    4066     892
iter:        13     164    4066     892     969
iter:        14     164     892     969    2914
iter:        15     892     969    2914    4056
iter:        16     969    2914    4056     978
iter:        17     969    4056     978     616
```

Fig. 3.6. Printing the progress of calculations

At the end, a final report, containing used settings and results of calculations, will be shown in the console. The same data is always saved in the TheName_XX.txt file in the current results subdirectory (Fig. 3.7). The sign XX gives the information about the ISP location (Fig. 3.1).

If one of the default 9 ISPs is used, the results are saved in the “shift_XX_YY” subfolder in TheName directory. When ISP is defined in a file, the subfolder is named “isp_ff”. Here for every case the same “isp_ff” subdirectory is used – if ISP coordinates in the file are changed, the results in this subfolder will be overwritten.

```
------------------------------------------------------------------
| PathFinder code - Copyright (C) Wojciech Sobieski 2009-2016    |
------------------------------------------------------------------

 Final report (the bed):

 domain geomoetry                  :  cuboids
 X-axis range               [m]    :        -0.000143        0.079332
 Y-axis range               [m]    :        -0.000124        0.079291
 Z-axis range               [m]    :        -0.000097        0.158477
 bed size in X-direction    [m]    :         0.079475
 bed size in Y-direction    [m]    :         0.079415
 bed size in Z-direction    [m]    :         0.158574
 number of particles        [-]    :     5000
 min. diameter              [m]    :         0.005897
 max. diameter              [m]    :         0.006266
 average diameter           [m]    :         0.006072

 Final report (settings):

 location of the Initial Starting Point (ISP)      :       00
 number of particles rejected from the top surface :        0
 method for calculating the triangle center        :       gc
 characteristic dimension for the Ideal Location   :    l_ave
 correction method for the Ideal Location (IL)     : function
 correction coefficient for the constant method    :   0.5000
 function parameter a for the function method      :   8.0000
 function parameter b for the function method      :   1.3000
 correction based on critical area of the triangle :      Yes
 critical normalized area of the triangle          :     3.00
 use the smoothing algorithm                       :      Yes
 function parameter a for the smoothing function   :   0.0700
 function parameter b for the smoothing function   :  90.0000

 Final report (results):

 number of path points                  [-]        :       84
 number of path rejected points         [-]        :        0
 number of corrected trangles           [-]        :        0

 bed height                             [m]        :         0.158570
 length of the path                     [m]        :         0.189968
 length of the path after smoothing     [m]        :         0.183787
 tortuosity                             [m/m]      :         1.198012
 tortuosity after smoothing             [m/m]      :         1.159032
 average angle between path sections    [^o]       :       142.084619

 inner surface of the solid body        [m^2]      :         0.579236
 specific surface of the porous body (K)[1/m]      :       578.764248
 specific surface of the porous body {C}[1/m]      :       987.953544

 volume of the bed                      [m^3]      :         0.001001
 volume of the solid part               [m^3]      :         0.000586
 volume of the porous body              [m^3]      :         0.000415
 porosity                               [m^3/m^3]  :         0.414179
 packing coefficient                    [m^3/m^3]  :         0.585821

 Ergun A                                [1/m^2]    :  19649597.005455
 Ergun 2*B                              [1/m]      :      4752.450777
 Kozeny-Carman (C_KC=1)                 [1/m^2]    :   6766473.535653

 The time of calculations               [s]        :        20.982134
```

Fig. 3.7. A final report (YADE example)

Depending on options in the TheName.dat file, visualizations in Gnuplot are available or not. In order to work correctly, the Gnuplot environment must be installed in the OS. Make sure, that the PLT extension is associated with the Gnuplot. In other case, the basic visualization won't be available.

3.4 The basic visualization in Gnuplot environment

During calculations, following drawings/figures/diagrams may be shown (in order of appearing):

- the diameters of all particles in the bed – function $d_i(n_i)$ (Fig. 3.8);
- distribution of the particle diameters – diagram showing how many particles have the same diameter (Fig. 3.9);
- the drawing of the current tetrahedron (not in a scale) (Fig. 3.10);
- the final path (not in a scale) (Fig. 3.11) - in the figure, the current path, as well as centres of spheres surrounding this path, are visible;
- the distribution of the triangle areas forming tetrahedrons basis (Fig. 3.12) - the greater this value, the more irregular arrangement of the particles is;
- the distribution of the triangle perimeters forming tetrahedrons basis (Fig. 3.13) - the greater the value, the more irregular arrangement of the particles is;
- the distribution of value of correction coefficient or correction function for all tetrahedrons (Fig. 3.14) - the smaller is this value, the more irregular shape has the basis of the current tetrahedron;
- the distribution of the quality I_F indicator for all tetrahedrons (Fig. 3.15) - the greater the deviation of this value, the worse is quality of predicted location of fourth particle forming the tetrahedral structure;
- the distribution of angles between triangle normals and Z-axis for all tetrahedrons (Fig. 3.16).

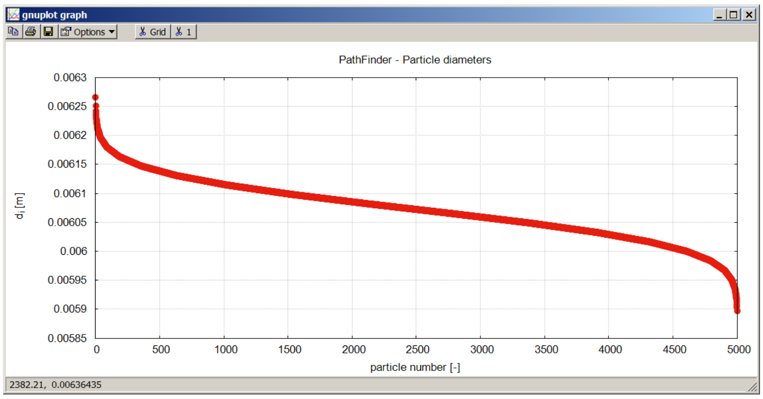


Fig. 3.8. The function $d_i(n_i)$

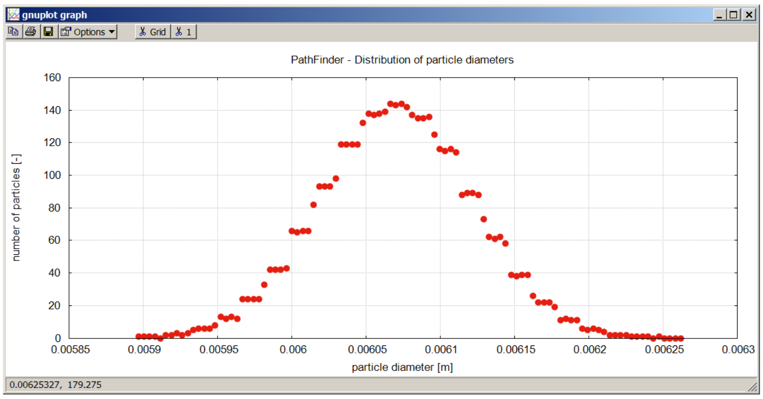


Fig. 3.9. The histogram of distribution of the particle diameters

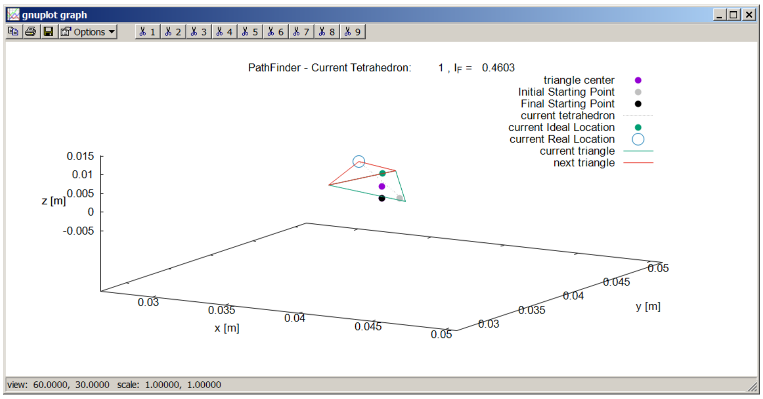


Fig. 3.10. The visualization of the current (here no 1) tetrahedron

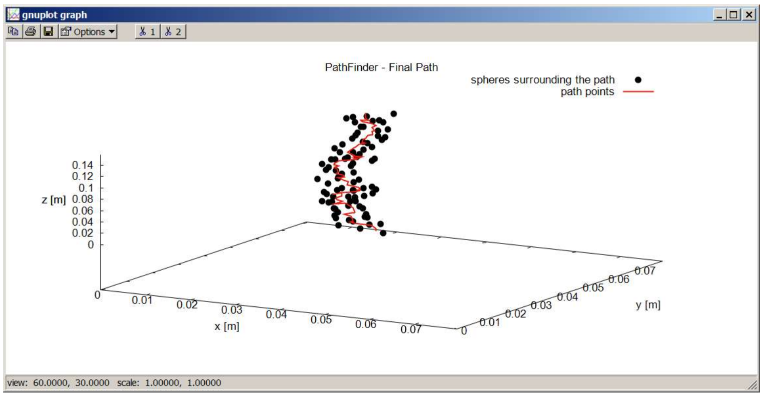


Fig. 3.11. The visualization of the final path

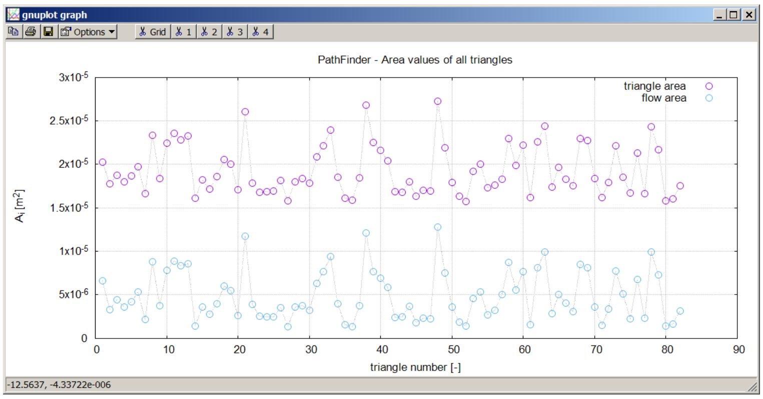


Fig. 3.12. The distribution of the triangle areas forming tetrahedrons basis

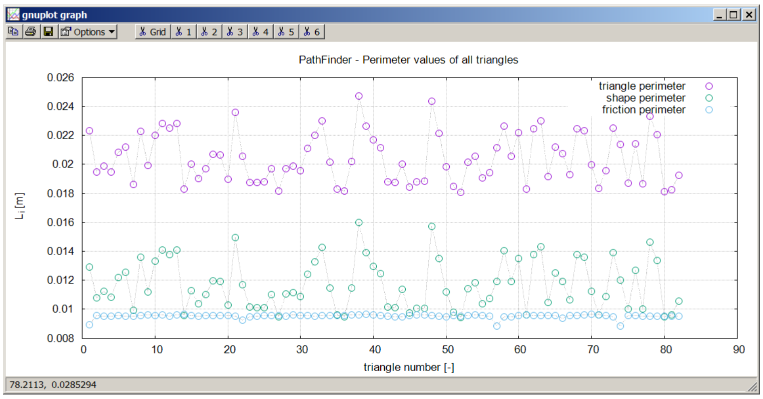


Fig. 3.13. The distribution of the triangle perimeters forming tetrahedrons basis

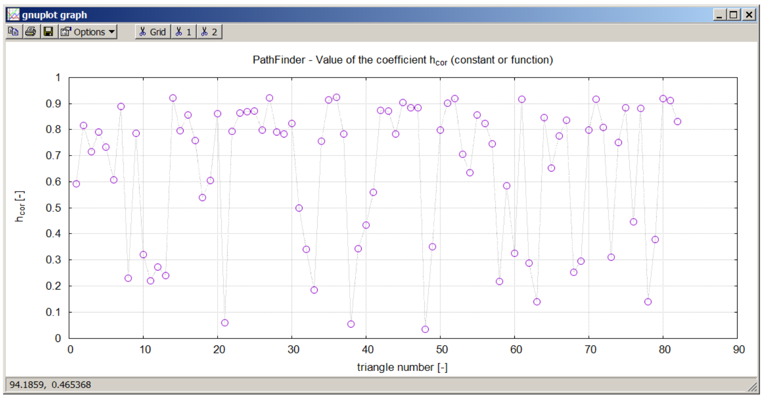


Fig. 3.14. The distribution of value of correction coefficient or correction function for all tetrahedron

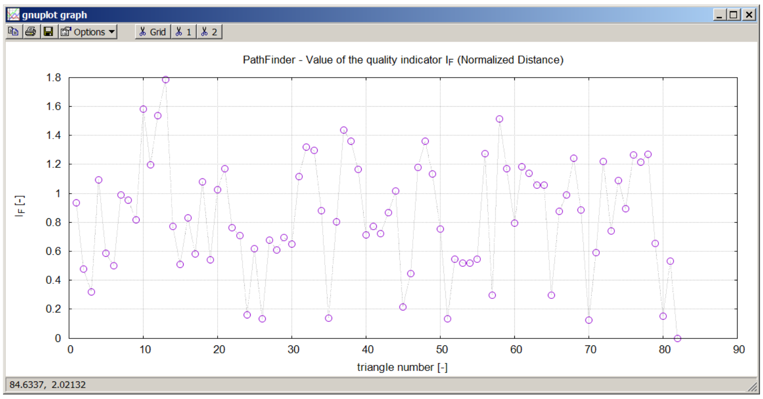


Fig. 3.15. The distribution of the quality I_F indicator for all tetrahedrons

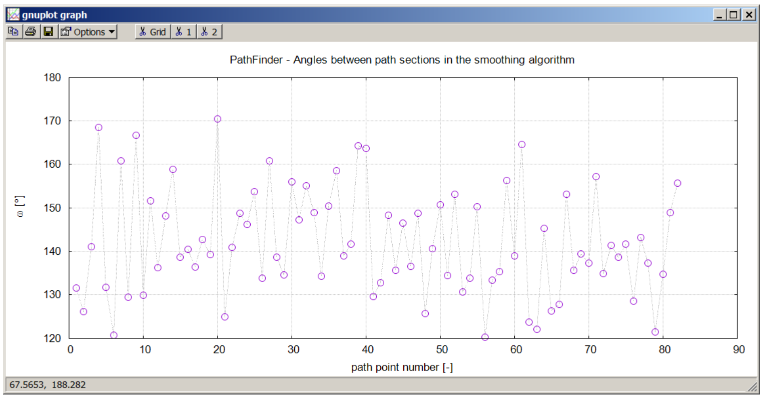


Fig. 3.16. The distribution of the angle between triangle normals and the Z-axis for all tetrahedrons

By default only the final path is shown.

All PLT scripts may be run later (apart from the script Plot_current.plt), but only results of the last calculation are available. If all 9 paths are calculated in one loop (the ISP option is equal to “al”), the path from ISP equal to “00” is calculated as the last. Needed files are described in Table 3.1.

Sometimes in Linux OS the graphical GUI of Gnuplot is not available by default. In this case type in the console:

```
sudo apt-get install gnuplot-x11
```

To connect the PLT scripts with the Gnuplot environment use the Right Mouse Button and define which application should open the chosen kind of files by default.

3.5 The Comparison folder

In Comparison folder contains following files:

- "Comparison.txt" – a file in which part of the results is collected. A new line with results is added for every PathFinder calculation. The file must be deleted or moved to start a new set of results. This file contains: the characteristic dimension, the method of predicting the Ideal Location, the number of spheres, the number of spheres rejected from the top surface, the number of path points, the number of rejected path points (duplicates), the average quality indicator, the number of corrected triangles, the tortuosity, the tortuosity after smoothing, and the ISP location.
- "Comparison_isp.txt" – a file in which part of the results is collected. A new line with results is added for every PathFinder calculation. The file must be deleted or moved to start a new set of results. This file contains: TheName of the current example, the ISP location, X, Y and Z coordinates of the first point in the path (the Final Starting Point), X, Y and Z coordinates of the last point in the path, the number of path points, the tortuosity, the maximum point deviation from the vertical line started from the ISP, and the point deviation for the last point in the path. This file may be used for visualisation of the tortuosity distribution for all used ISPs (Fig. 3.19). This file is used for saving data for different approaches foor ISPs choices, such as: 00, +-, 0-, --, -0, -+, 0+, ++, "fc" and "fp".
- "Comparison_rg.txt" – a file with the same data like in the previous file, but only for the Regular Grid Method.
- A set of files with ".pf" extension, given in subdirectories ("isp_fc", "isp_fp", "isp_rg" and "shift_XX_YY", where XX and YY denotes signs 00, +-, 0-, --, -0, -+, 0+, ++, see Point 3.3). In this file, three columns for every path are collected: the Z coordinate of the current path point, the distance between the current point and his neighbouring point located below, and the distance between all path points and the vertical line started from the ISP. These files may be used for visualisation of the deviation of path points location from a vertical line started from FSP (Fig. 3.21) or for visualisation of the distance between adjacent path points in Z direction (Fig. 3.22)
- "d-e-tau-s0.pf" – a file with average diameter of all particles in the bed, porosity, tortuosity and values of the specific surface in Kozeny and Carman definition;
- "e_d_a_E.pf" – a file with porosity, average diameter of all particles in the bed and value of A term of Ergun equation;
- "e_d_b_E.pf" – a file with porosity, average diameter of all particles in the bed and value of B term of Ergun equation;
- "e_s0_a_KC.pf"– a file with porosity, values of specific surface in Kozeny and Carman definition and A term of Kozeny-Carman equation.
- "e_tau_a_KC.pf" – a file with the porosity, the tortuosity and A term of Kozeny-Carman equation.

Last 5 files may by used by additional software described in the Point 7.8.

In the Comparison directory, four Gnuplot scripts are created automatically:

- "Comp_np.plt" – Gnuplot script for comparing numbers of path points in different paths (Fig. 3.17).

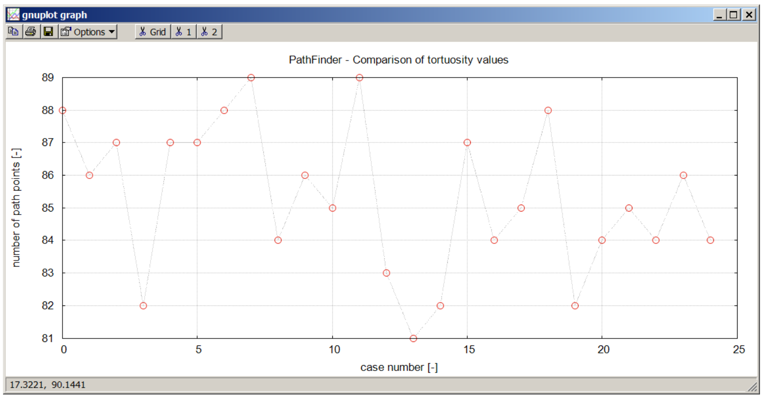


Fig. 3.17. Comparison of the number of path points (the Standard Method was used here)

- "Comp_tau.plt" – Gnuplot script for comparing values of tortuosity obtained for different IPSs (Fig. 3.18).

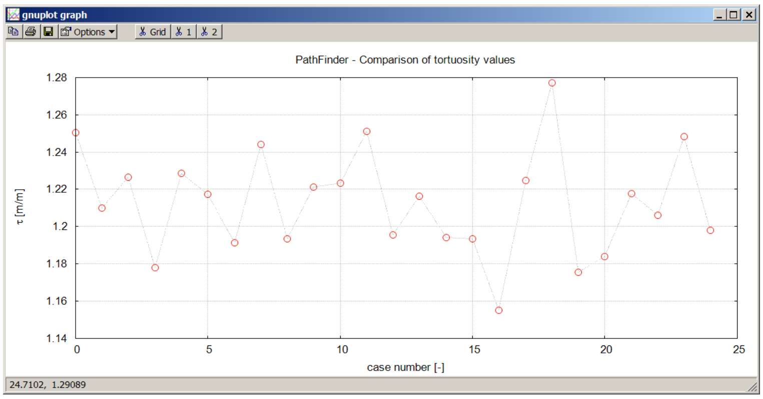


Fig. 3.18. Comparison of the tortuosity values (the Standard Method was used here)

- "Comp_tau_isp.plt" – Gnuplot script for visualization of the tortuosity distribution in all used ISPs (Fig. 3.19), except from values obtained with the use of the Regular Grid Method.

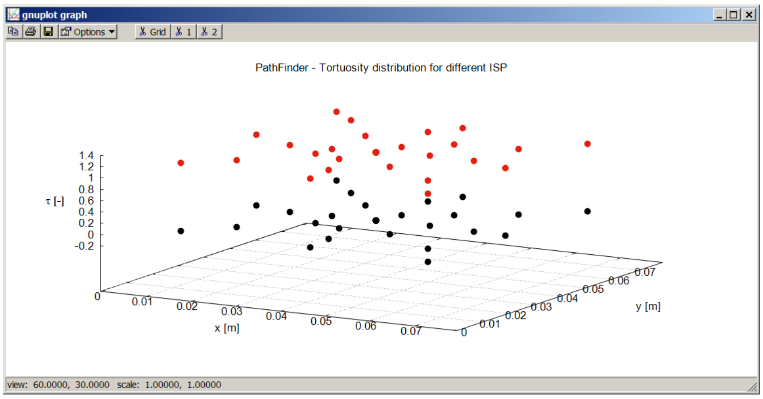


Fig. 3.19. The tortuosity distribution in function of the ISP location

- "Comp_tau_rg.plt" – Gnuplot script for visualization of the tortuosity distribution in all used ISPs using the Regular Grid Method (Fig. 3.20).

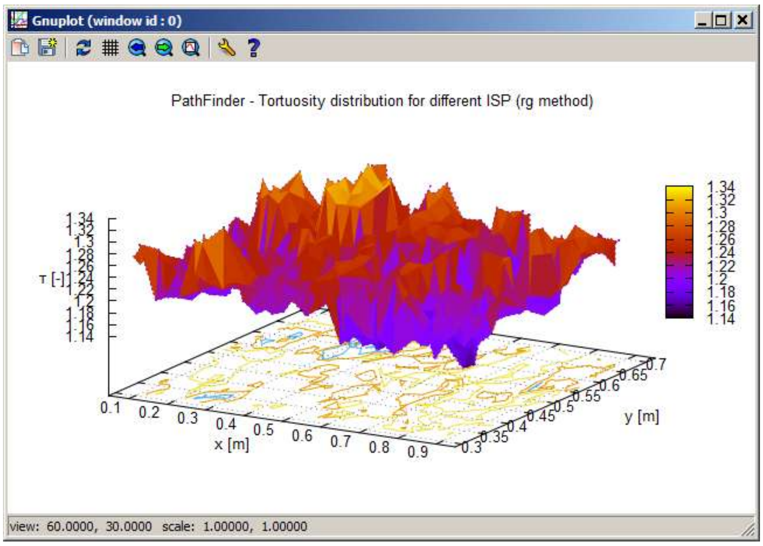


Fig. 3.20. The tortuosity field obtained with the use of the Regular Grid Method

In the Comparison folder, additional subfolder exists, in which files related to the deviation analysis are collected. The name of this subfolder depends on the approach to ISP choice and it may be named: "isp_fc", "isp_fp", "isp_rg" or "shift_XX_YY". Files with the "pf" extension contain: the Z coordinate of the current path point, the distance in the Z direction between the current and the previously path point, the deviation, and the non-dimensional deviation.

In the same subfolder, two Gnuplot scripts are created:

- Comp_dev.plt – Gnuplot script for visualisation of the distance between all path points and the vertical line starting from the ISP (Fig. 3.21).
- Comp_dz.plt – Gnuplot script for visualisation of the distance between adjacent path points in Z direction (Fig. 3.22).

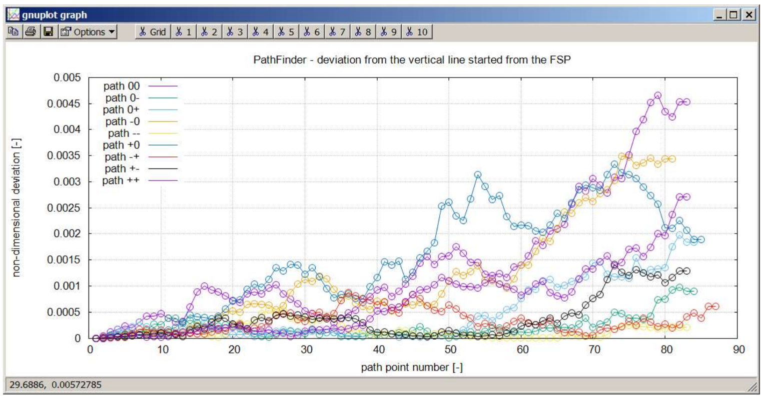


Fig. 3.21. The deviation of path points

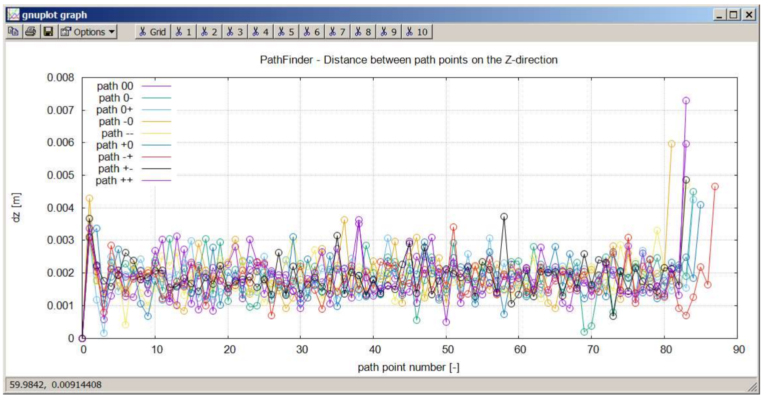


Fig. 3.22. The distance between adjacent path points

Table 3.1: Files used in scripts by the Gnuplot environment

Script name	Used files	Description	Available after calculations
Plot_area.plt	TheName_XX.res (1:2)	scalars for the tetrahedral structures	yes
Plot_current	TheName.in	the input data	yes
	TheName_XX.pp	points of path	yes
	TheName.isp	the location of the Initial Starting Point	no
	TheName.fsp	the location of the Final Starting Point	no
	TheName.ctr	the current triangle	no
	TheName.cth	the current tetrahedron	no
	TheName.4il	the current Ideal Location	no
	TheName.4rl	the current Real Location	no
	TheName.ntr	the next triangle	no
Plot_d.plt	TheName.in	the input data	yes
Plot_dis_d.plt	TheName.dis	diameter distribution	yes
Plot_final.plt	TheName_XX.sp	the spheres surrounding the path	yes
	TheName_XX.pp	points of path	yes
Plot_if.plt	TheName_XX.res (1:5)	scalars for the tetrahedral structures	yes
Plot_il_cor.plt	TheName_XX.res (1:4)	scalars for the tetrahedral structures	yes
Plot_normal.plt	TheName_XX.res (1:6)	scalars for the tetrahedral structures	yes
Plot_omega.plt	TheName_XX.res (1:7)	scalars for the tetrahedral structures	yes
Plot_omega_cor.plt	TheName_XX.res (1:8)	scalars for the tetrahedral structures	yes
Plot_perimeter.plt	TheName_XX.res (1:3)	scalars for the tetrahedral structures	yes

4 PathGUI overlay

4.1 General description

PathGUI graphical overlay is the element of the PathFinder project and it mainly serves for the creation of the configuration files. In addition, it allows to run PathFinder code (via a shell script), scripts of Gnuplot environment and ParaView program, designed for three-dimensional visualization of the results.

PthGUI program was written in a visual programming environment Lazarus[7], based on Free Pascal language; it can be used on Windows or Unix\Linux. In the case of Unix\Linux, User should ensure that the program has the right to execute. In the case of copying the compiled program, such a right should be given manually (e.g. through system console, by giving chmod instruction). In the case of compiling the source code, the right to execute is assigned automatically by the operating system.

4.2 Handling tasks

Overlay window consists of a main menu, a toolbar, and five tabs.

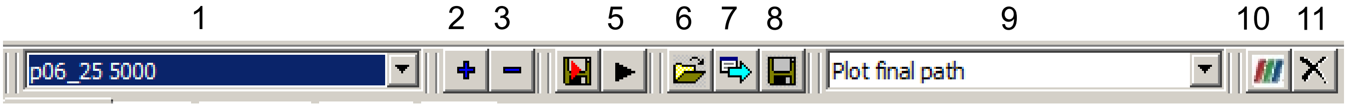


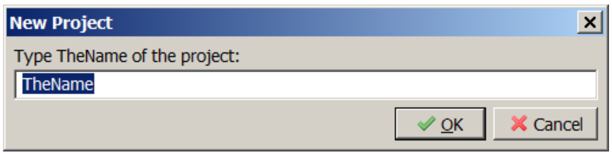
Fig. 4.1. View of a toolbar of a pathGUI overlay

Meaning of the elements of the toolbar is as follows (Fig. 4.1):

1. List of tasks – contains names of all defined tasks. These tasks must be in the same directory (hereinafter refferred to as the root directory), in which PathFinder program is, as well as pathGUI overlay. The list of tasks is stored in a text file named "pathGUI.tn" (if necessary, this file can be edited manually).
2. Button of adding task to the list. After clicking on it, you will see two windows with questions on the name of the task (Fig. 4.2) and on the number of spheres in a bed (Fig. 4.3). After approval of answers on both questions, in the root folder task folder is created, as well as the shell script used to run the PathFinder program. In the task folder, a configuration file containing default settings, will be also created. Because the program does not detect the shape of the deposit, User will be asked to choose the type of computational domain (rectangular or cylindrical).
3. Button deleting tasks from the list. After clicking it, from the list of tasks ("pathGUI.tn" file) will be deleted task, whose name is displayed in the task list. All files related to the selected task will remain unchanged. Deleted task can add again to the list of tasks at any time.
4. Button saving the current configuration of the task to the „TheName.dat” file.
5. Button of task start. Clicking the button launches the shell script and starts the Pathfinder program. For Unix/Linux User should select "GUI Settings" in the overlay menu and enter the name of the shell together with the relevant calling arguments (if needed). For example, in case of a xUubuntu system, it is a "xfce4-terminal -x" command. Instruction is stored in the "pathGUI.s" file.
6. Button loading the configuration file. This key allows you to copy settings from another task.
7. Button of default settings. Please be aware that after using it, User should check and, if necessary, change the type of computational domain.

[7] Lazarus Home Page [on-line]. URL: http://www.lazarus-ide.org/

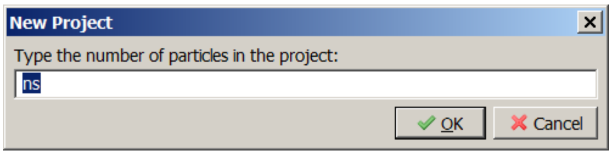
8. Button saving the current configuration of the task into any configuration file. After clicking on it, appears the standard for the used operating system window for saving files.
9. List of Gnuplot scripts. It allows running the chosen Gnuplot script for recently calculated path. Scripts will launch automatically, provided that the "*.plt" files have been previously associated with the Gnuplot environment.
10. Button for starting the Paraview environment. On Windows, its correct operation requires providing a path to this environment. It can be done through the "GUI Settings" command in the main menu. The path is stored in the "pathGUI.s" file.
11. Button for exiting the program.


New Project

Type TheName of the project:

TheName

Fig. 4.2. The pathGUI tool – input box to enter the project name


New Project

Type the number of particles in the project:

ns

Fig. 4.3. The pathGUI tool – input box to enter the number of particles

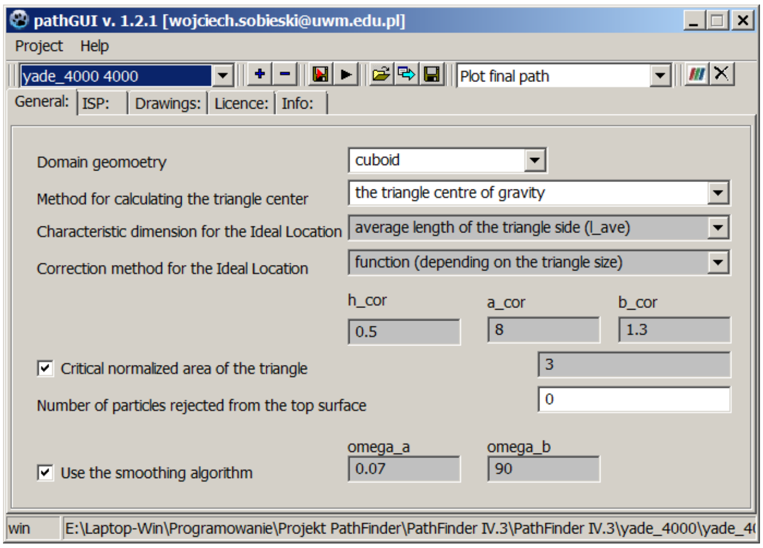
4.3 Configuration tabs

Basic settings for tasks are defined in the pathGUI overlay on the General tab (Fig. 4.4). Meaning of of particular items is as follows:

- Domain geometry – the shape of computational space. The PathFinder program allows analysis beds of rectangular or cylindrical topology. For "p06_25" example, deposit has rectangular shape (see Fig. 6.1), what results from the way of its generation with the use of Discrete Element Method in YADE package.
- Method for calculating the triangle center – method of calculating coordinates representing the center of the triangle formed by the spheres belonging to the base of the tetrahedral structure. There are two possible options:
  - A characteristic point is the geometric center of the triangle.
  - A characteristic point is the center of gravity of the outline of the pore space lying in the triangle plane (the recommended option, added in 2014 [4]).
- Characteristic dimension for the Ideal Location – characteristic dimension needed for the estimation of earlier described Ideal Location.
- Vertex coordinates of tetrahedral structure are calculated assuming that analyzed tetrahedron is a regular tetrahedron with a side length equal to the characteristic dimension. There are two possible options:
  - The characteristic dimension is the average particle diameter (option of historical significance, not recommended).
  - The characteristic dimension is the average length of the side of the triangle forming the basis for the tetrahedral structure (recommended).
- Correction method for the Ideal Location – method of correcting the vertex distance from the base surface of tetrahedral structure. Previous studies [18] have shown that the quality of search (determined on the basis of a special index) is better, if the Ideal Location is closer to the base surface of tetrahedral structure than that it would result from the geometry of the

regular tetrahedron. This is because in actual beds, particles do not form a perfectly rhombic topology: the particles forming the base are often slightly shifted from each other, and so the fourth particle may partially “slide” between them. Reducing the distance between the surface of the base surface of tetrahedral structure and Ideal Location can be made by a constant value (0.5) or by the function based on the distance between the spheres forming the basis of the tetrahedral structure. Function coefficients (8.0 and 1.3) have been determined on the basis of previous studies [18] and should not be changed.

- Critical normalized area of the triangle. Previous studies [18] have shown that if for spheres having the same diameters, area of the triangle formed by the non-contacted spheres is three times larger than the area formed by spheres in contact, the fourth sphere may be wholly contained inside (between the spheres forming the basis for the tetrahedral structure). In the case of high porosity beds, or high variance of particle size distribution, this value can be increased.
- Number of particles rejected from the top surface – the number of elements of the set, counting from the end of the set (i.e. from the spheres located highest in the main direction) that will be rejected from the calculations. This option can be used in situations, in which the upper bed surface is not flat (so there are more cavities than in other regions of the bed) - in these cases the bed porosity is overestimated.
- Use the smoothing algorythm. Selecting this option causes calculation of the additional value of tortuosity, with the use of the function rounding connections for individual sections of the path. This option is of historical importance and currently is not recommended (its selection does not affect any of other results in any way).



Domain geomoetry: cuboid

Method for calculating the triangle center: the triangle centre of gravity

Characteristic dimension for the Ideal Location: average length of the triangle side (l_ave)

Correction method for the Ideal Location: function (depending on the triangle size)

h_cor: 0.5 a_cor: 8 b_cor: 1.3

☑ Critical normalized area of the triangle: 3

Number of particles rejected from the top surface: 0

☑ Use the smoothing algorithm omega_a: 0.07 omega_b: 90

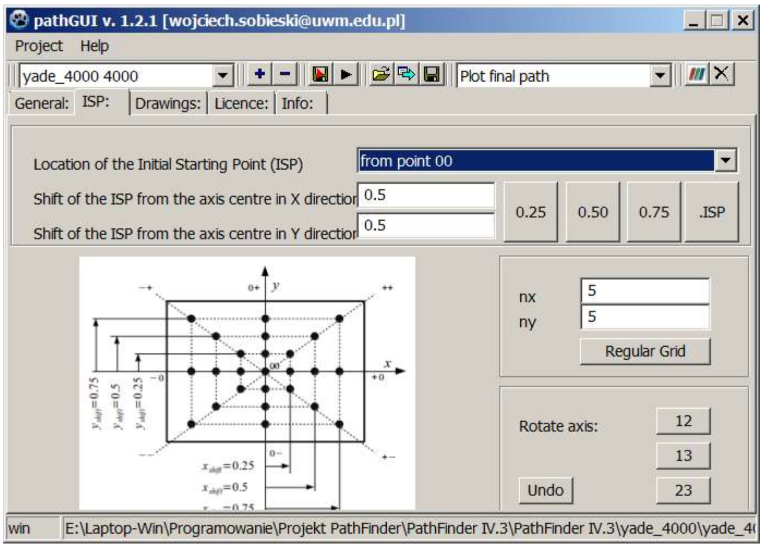
win E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4(

Fig. 4.4. View of a pathGUI overlay window and General tab

The second tab of pathGUI overlay concerns a method of selecting Initial Starting Points (description was given in Section3.3).

In order to facilitate the ISPs selection, on the tab a scheme of starting points is shown for the selected geometry: rectangular (Fig. 4.5) or cylindrical (Fig. 4.6). Schemes are also shown for different methods of ISP selection (Figs. 4.7, 4.8 and 4.9).

On the ISP tab is also available function for changing the position of the bed, relative to the adopted coordinate system. It is performed by changing the columns containing the coordinates of the spheres in "TheName.in" file. The numbers on the buttons indicate which columns will be replaced. This option is very useful when you want to analyze paths in other directions than originally assumed main direction. During the column replacement, the backup copy of the previous version is created and it can be recovered with the Undo button.



pathGUI v. 1.2.1 [wojciech.sobieski@uwm.edu.pl]

yade_4000 4000

Plot final path

Location of the Initial Starting Point (ISP): from point 00

Shift of the ISP from the axis centre in X direction: 0.5

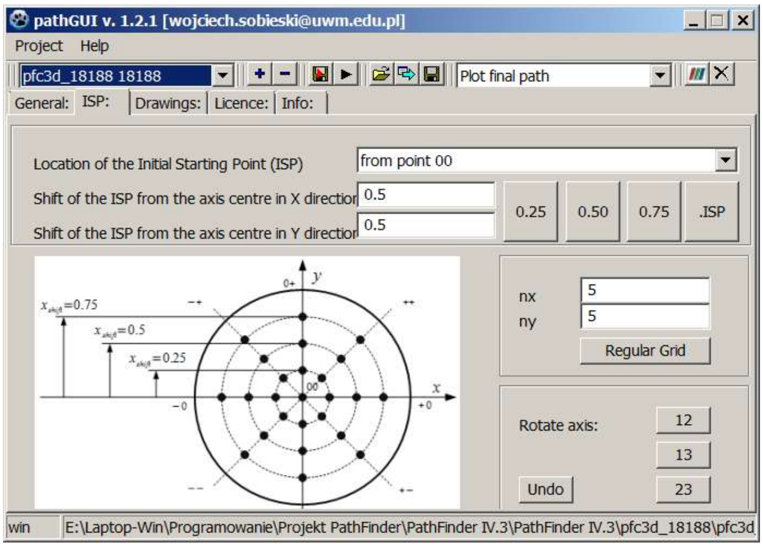
Shift of the ISP from the axis centre in Y direction: 0.5

nx: 5

ny: 5

win E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4(

Fig. 4.5. View of a pathGUI overlay window and ISP tab (rectangular domain)



pathGUI v. 1.2.1 [wojciech.sobieski@uwm.edu.pl]

pfc3d_18188 18188

Plot final path

Location of the Initial Starting Point (ISP): from point 00

Shift of the ISP from the axis centre in X direction: 0.5

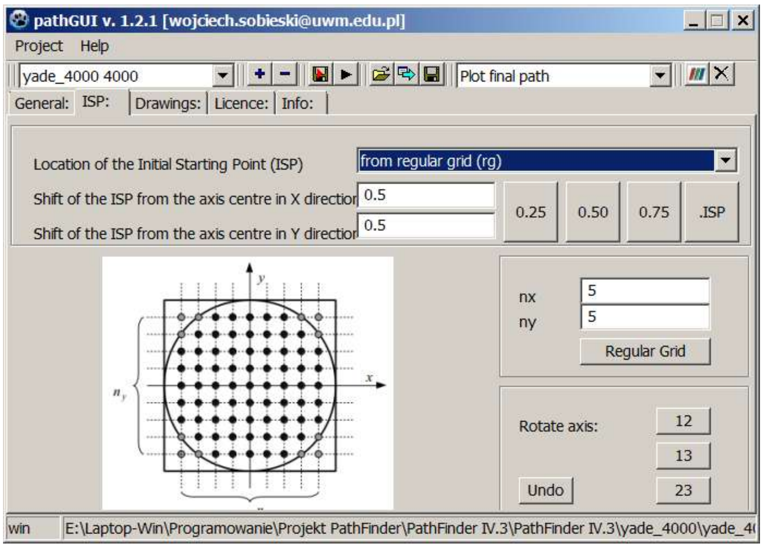
Shift of the ISP from the axis centre in Y direction: 0.5

nx: 5

ny: 5

win E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\pfc3d_18188\pfc3d

Fig. 4.6. View of a pathGUI overlay window and ISP tab (cylindrical domain)



yade_4000 4000

Location of the Initial Starting Point (ISP): from regular grid (rg)

Shift of the ISP from the axis centre in X directior: 0.5

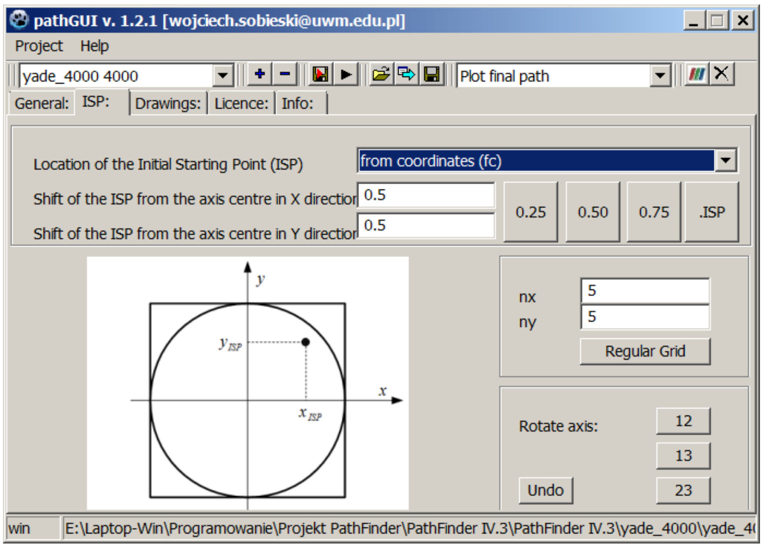
Shift of the ISP from the axis centre in Y directior: 0.5

nx: 5

ny: 5

win E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4(

Fig. 4.7. View of a pathGUI overlay window and ISP tab, including the ISPs scheme for Regular Grid Method



yade_4000 4000

Location of the Initial Starting Point (ISP): from coordinates (fc)

Shift of the ISP from the axis centre in X directior: 0.5

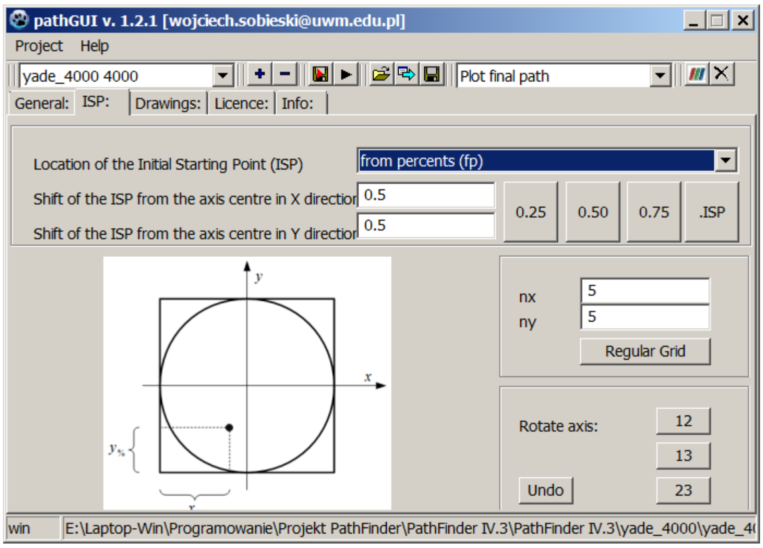
Shift of the ISP from the axis centre in Y directior: 0.5

nx: 5

ny: 5

win E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4(

Fig. 4.8. View of a pathGUI overlay window and ISP tab with idea of chosing ISP based on coordinates



yade_4000 4000

Location of the Initial Starting Point (ISP): from percents (fp)

Shift of the ISP from the axis centre in X directior: 0.5

Shift of the ISP from the axis centre in Y directior: 0.5

nx: 5

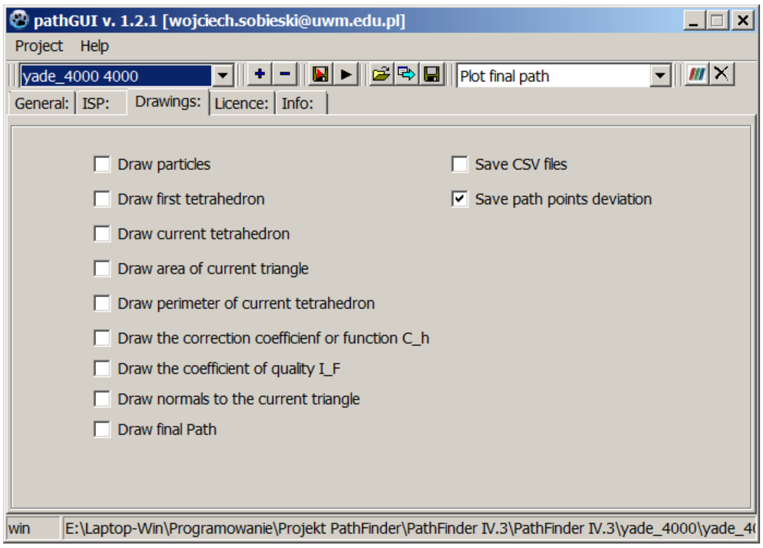
ny: 5

win E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4(

Fig. 4.9. View of a pathGUI overlay window and ISP tab with idea of chosing ISP based on percentage of axis range

On the Drawings tab (Fig. 4.10) plots can be selected, which are to be visualized during calculations. At default settings, charts visualization is disabled (except for the final drawing of the path, i.e. the "Draw Final Path" option), as they require continuous closing of Gnuplot environment windows, what is not convenient. Furthermore, these charts can be viewed later, by running the appropriate "*.plt" scripts (manually or using list of scripts, located on the toolbar). The exception is a drawing showing the current scheme of tetrahedral structure with characteristic points marked, because when the calculation is complete, only the last structure can be visualized. To analyze each structure sequentially, "Draw current tetrahedron" option has to be turned on, but keep in mind that these figures may be tens or hundreds, depending on the size of the deposit, and each of them will have to be manually closed later. For this reason, "Draw first tetrahedron" option is useful, as it allows visualizing the beginning of the path: the first tetrahedral structure with its starting points (initial and final). Drawings of tetrahedral structures are not written to disk, since their analysis usually requires performing rotation and scaling, what is not possible after saving it to the image file. Besides, saving complete data on each structure would occupy too much disk space.

On the Drawings tab there is also an option allowing to generate CSV files (by default this option is not checked). When selected, deposit, as well as points forming individual paths, are saved in this format.



□ Draw particles
□ Draw first tetrahedron
□ Draw current tetrahedron
□ Draw area of current triangle
□ Draw perimeter of current tetrahedron
□ Draw the correction coefficienf or function C_h
□ Draw the coefficient of quality I_F
□ Draw normals to the current triangle
□ Draw final Path

□ Save CSV files
☑ Save path points deviation

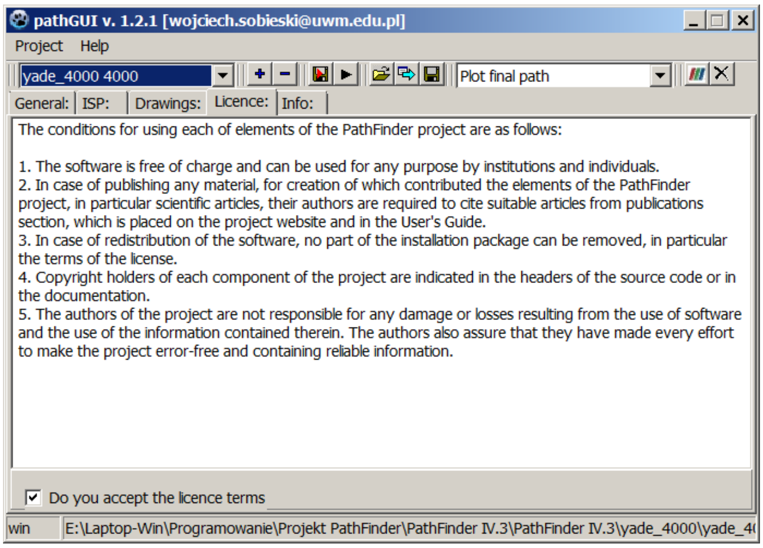
E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4(

Fig. 4.10. View of a pathGUI overlay window and Drawings tab

The last option is associated with saving files containing the so-called deviations of the path points. Deviation is understood as the distance of the current path point from the vertical line, going from the ISP. If you do not want to use this option, you can turn it off, because it causes generation of a large number of small files (especially in Regular Grid Method).

The next tab is Licence tab (Fig. 4.11), where you can see the terms and conditions of the program. There is also a check box that you have to check in order to legally use the program. Without this, the program will not perform calculations.

The last tab of pathGUI overlay is of informative character. It includes contact to the author and his affiliation (Fig. 6.14).

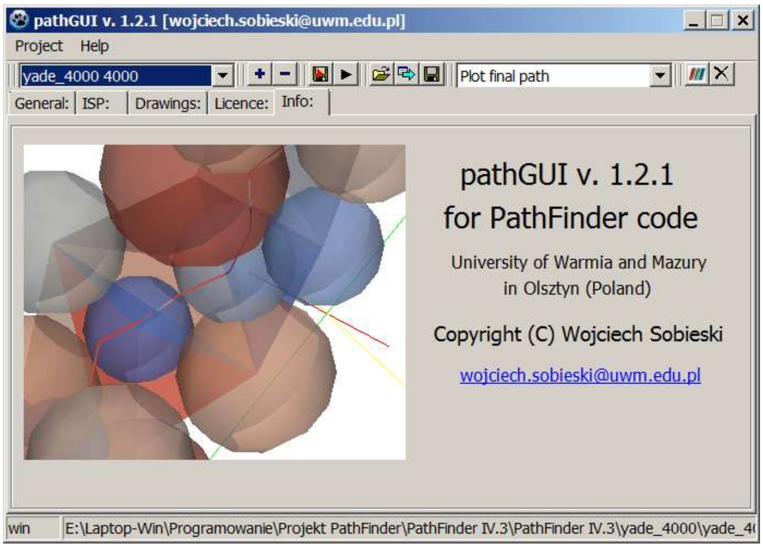

pathGUI v. 1.2.1 [wojciech.sobieski@uwm.edu.pl]
Project
Help
yade_4000 4000
Plot final path
General:
ISP:
Drawings:
Licence:
Info:
win
E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4

The conditions for using each of elements of the PathFinder project are as follows:

1. The software is free of charge and can be used for any purpose by institutions and individuals.
2. In case of publishing any material, for creation of which contributed the elements of the PathFinder project, in particular scientific articles, their authors are required to cite suitable articles from publications section, which is placed on the project website and in the User's Guide.
3. In case of redistribution of the software, no part of the installation package can be removed, in particular the terms of the license.
4. Copyright holders of each component of the project are indicated in the headers of the source code or in the documentation.
5. The authors of the project are not responsible for any damage or losses resulting from the use of software and the use of the information contained therein. The authors also assure that they have made every effort to make the project error-free and containing reliable information.

☑ Do you accept the licence terms

Fig. 4.11. View of a pathGUI overlay window and Licence tab


pathGUI v. 1.2.1 [wojciech.sobieski@uwm.edu.pl]
Project
Help
yade_4000 4000
Plot final path
General:
ISP:
Drawings:
Licence:
Info:
win
E:\Laptop-Win\Programowanie\Projekt PathFinder\PathFinder IV.3\PathFinder IV.3\yade_4000\yade_4

pathGUI v. 1.2.1
for PathFinder code

University of Warmia and Mazury
in Olsztyn (Poland)

Copyright (C) Wojciech Sobieski

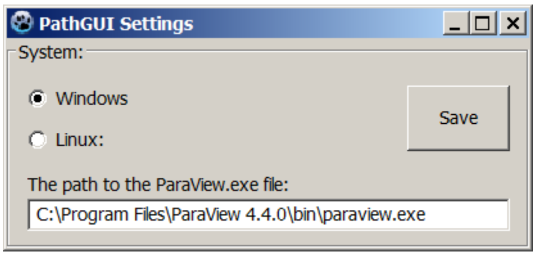
wojciech.sobieski@uwm.edu.pl

Fig. 4.12. View of a pathGUI overlay window and Info tab

Concluding a description of pathGUI overlay, the main menu should be mentioned. It contains two groups of options: Project, in which only the "GUI Settings" option is not a duplication of the features available from the toolbar, and Help, in which there are references to the User's Guide in PDF format, to the author website and to the PathFinder project website.

In the PathGUI, the current operating system may be chosen (Fig. 4.13). After confirmation, the PathFinder.sys file is saved. This is a file needed for correct compilation of the PathFinder source code. In the same place the name of default shell should be defined (in Unix/Linux) (Fig. 4.13) or the location of the ParaView executable file (in Windows) (Fig. 4.14). The name of default shell (with call parameters if needed) is used in Linux for running SH scripts and other executable files.

The Window with the current operating system and the Linux terminal may be called in main menu: Project – GUI Settings.



System:

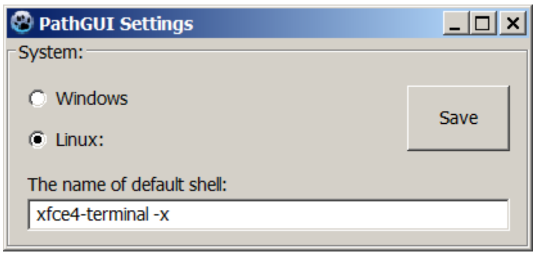
● Windows

○ Linux:

The path to the ParaView.exe file:

C:\Program Files\ParaView 4.4.0\bin\paraview.exe

Fig. 4.13. The pathGUI tool – definition of the current OS and the path to the ParaView executable file



System:

○ Windows

● Linux:

The name of default shell:

xfce4-terminal -x

Fig. 4.14. The pathGUI tool – definition of the current OS and the terminal name for Linux

5 The visualization in ParaView software

5.1 The visualization of the bed using VTK files

This kind of visualization shows all spheres of the bed in their actual size.

The visualization of the bed may be done through following steps:

- Run the ParaView software;
- Open TheName.in.vtk file;
- Press the Apply button;
- Turn the Glyph filter ;
- Set options for the Glyph filter:
  - Glyph type: Sphere;
  - Scale mode: Scalar;
  - Set scalar factor: 1;
- Press the Apply button;
- In the Pipeline Browser, change the "Glyph1" name to "The bed" (Fig. 5.1).

To see the inside of the bed, reduce the value of the Capacity option.

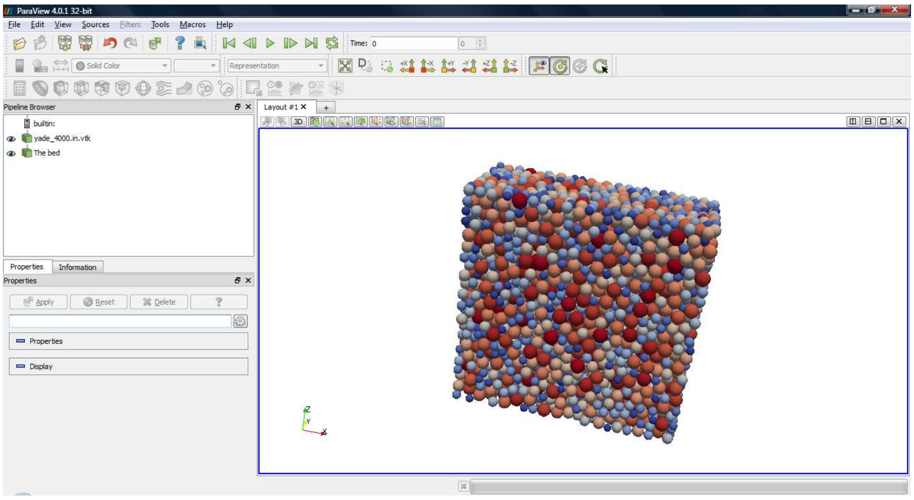


Fig. 5.1. Visualization of the bed

5.2 The visualization of the bed using CSV files

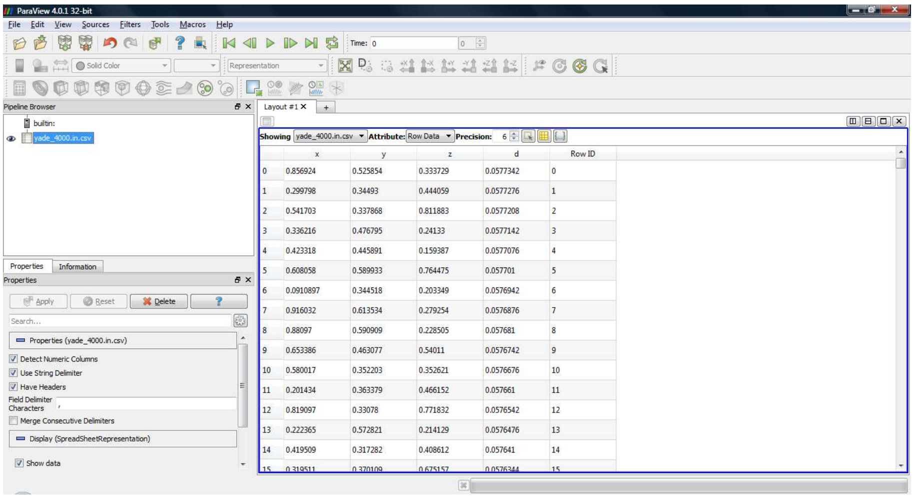
The option save_csv in the setting file (TheName.dat) must have the 'Y' value.

This kind of visualization shows not the spheres, but only the centres of spheres in the bed.

The visualization of the bed may be done through following steps:

- Run the ParaView software;
- Open TheName.in.csv file and confirm it with the Apply button (Fig. 5.2);
- Run “Table to Points” filter from the main menu Filters → Alphabetical;
- Determine the meaning of each column in the Properties panel and confirm it with the Apply button (Fig. 5.3);

- Close the main graphical window with the icon ☒ in the right top corner and create new 3D View (Fig. 5.4);
- Click the eye 👁 in the Pipeline Browser located left to the current TableToPoints;
- In the Pipeline Browser, change the "TableToPoints" name to "The bed" (Fig. 5.5).



Time: 0

Properties (yade_4000.in.csv)

☑ Detect Numeric Columns
☑ Use String Delimiter
☑ Have Headers
Field Delimiter Characters ,
☐ Merge Consecutive Delimiters

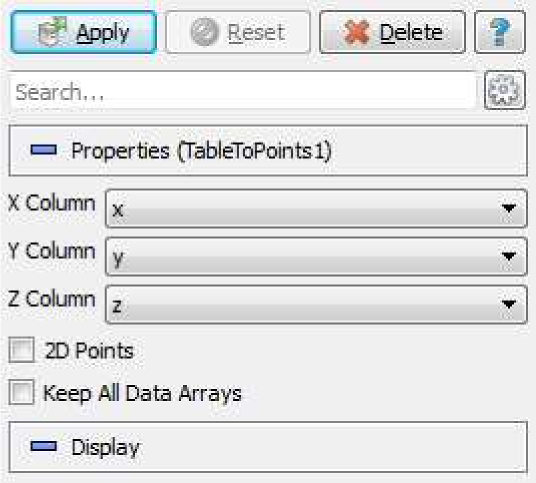
Display (SpreadSheetRepresentation)

☑ Show data

Showing yade_4000.in.csv Attribute: Row Data Precision: 6

	x	y	z	d	Row ID
0	0.856924	0.525854	0.333729	0.0577342	0
1	0.299798	0.34493	0.444059	0.0577276	1
2	0.541703	0.337868	0.811883	0.0577208	2
3	0.336216	0.476795	0.24133	0.0577142	3
4	0.423318	0.445891	0.159387	0.0577076	4
5	0.608058	0.589933	0.764475	0.057701	5
6	0.0910897	0.344518	0.203349	0.0576942	6
7	0.916032	0.613534	0.279254	0.0576876	7
8	0.88097	0.590909	0.228505	0.057681	8
9	0.653386	0.463077	0.54011	0.0576742	9
10	0.580017	0.352203	0.352621	0.0576676	10
11	0.201434	0.363379	0.466152	0.057661	11
12	0.819097	0.33078	0.771832	0.0576542	12
13	0.222365	0.572821	0.214129	0.0576476	13
14	0.419509	0.317282	0.408612	0.057641	14
15	0.319511	0.370109	0.675157	0.0576344	15

Fig. 5.2. Opening a CSV file for visualization of the bed



Properties (TableToPoints1)

X Column x
Y Column y
Z Column z
☐ 2D Points
☐ Keep All Data Arrays

Display

Fig. 5.3. Determinating the meaning of columns

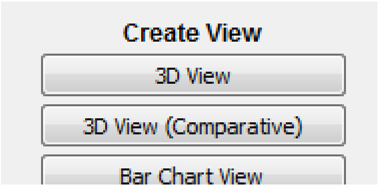
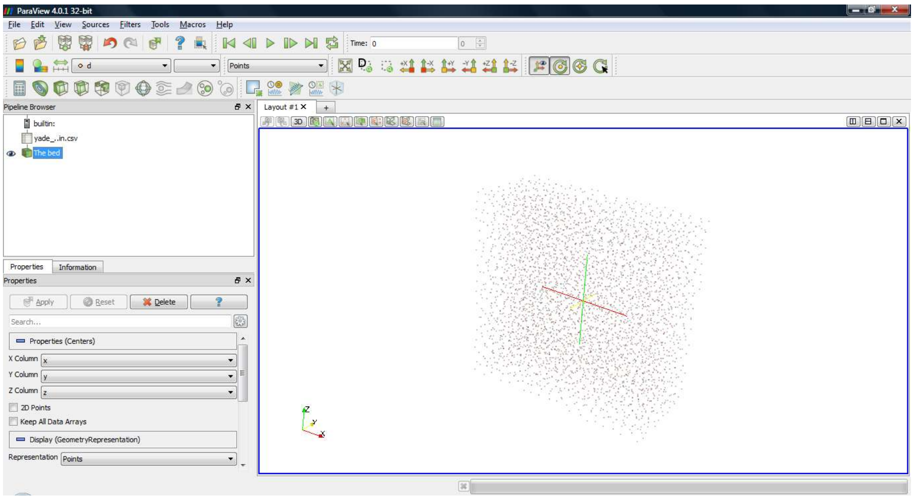


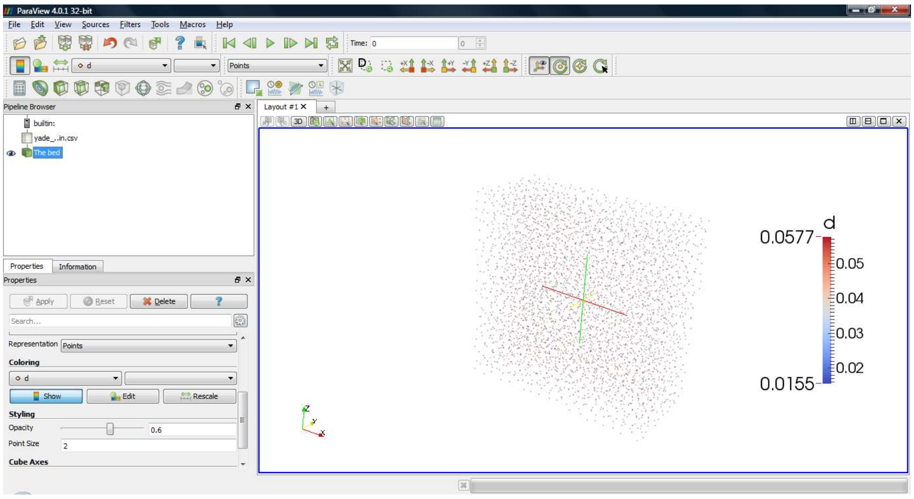
Fig. 5.4. Opening the 3D View (YADE example)

The sphere centres may be coloured by the particle diameter. To do this, the d parameter from Colouring list in the Properties panel must be chosen. A legend can be added to the drawings by clicking the Show button (Fig. 5.6).



Properties (Centers)
X Column x
Y Column y
Z Column z
□ 2D Points
□ Keep All Data Arrays
Display (GeometryRepresentation)
Representation Points

Fig. 5.5. Visualization of all sphere centres in the bed



Representation Points
Coloring
d
Styling
Opacity 0.6
Point Size 2
Cube Axes

Fig. 5.6. Visualization of all sphere centres in the bed coloured by particle diameters

5.3 The visualization of the path using VTK files

This kind of visualization shows the path as a polyline.

The visualization of the path may be done through following steps:

- Run the ParaView software;
- Open TheName_XX.pp.vtk file and confirm it with the Apply button;
- Set the colour of the polyline with the Edit button in the Colouring section.
- In the Pipeline Browser, change the "TheNameXX.pp.vtk" name to "Path XX" (Fig. 5.7).

In the same way next paths may be added to the drawing (Fig. 5.8).

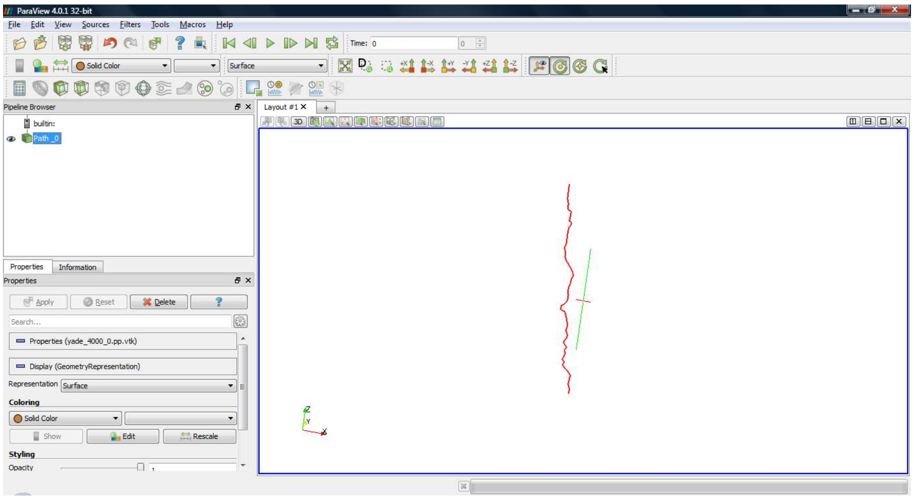


Fig. 5.7. Visualization of a path

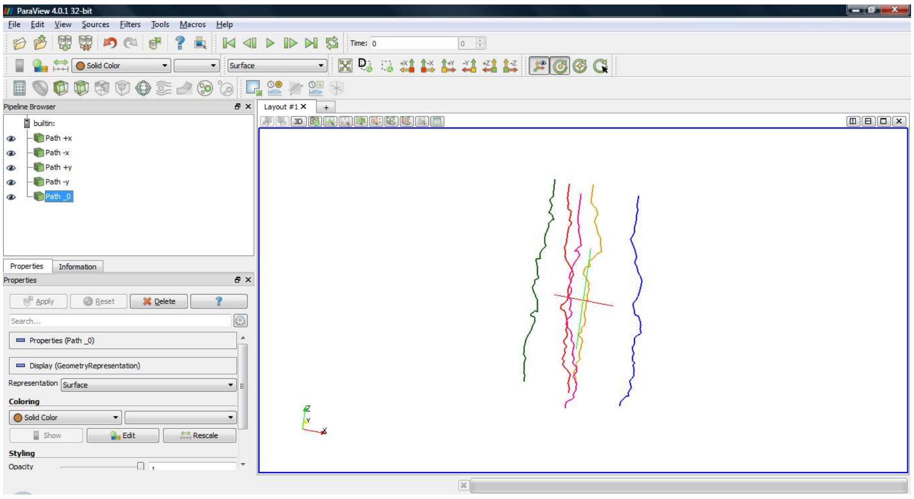


Fig. 5.8. Visualization of a set of paths

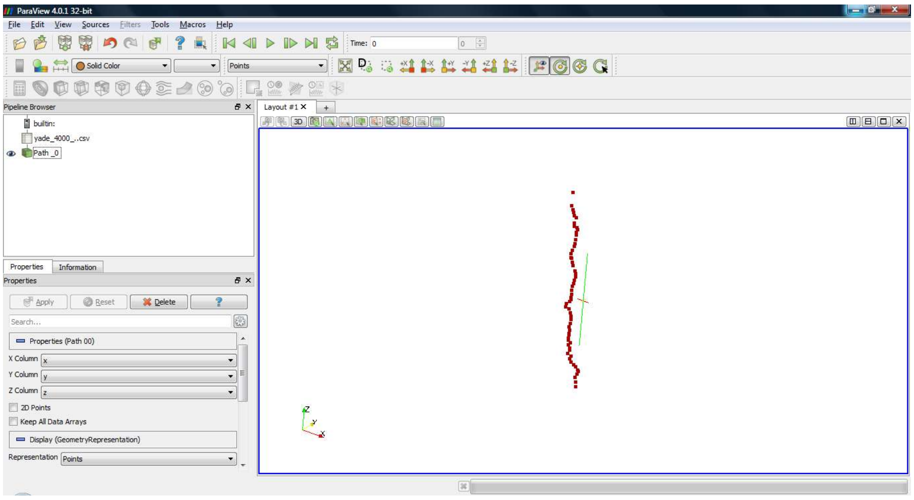
5.4 The visualization of the path using CSV files

The option save_csv in the setting file (TheName.dat) must have the 'Y' value.

This kind of visualization shows the path as a set of points.

The visualization of the path may be done through following steps:

- Run the ParaView software;
- Open TheNameXX.csv file and confirm it with the Apply button;
- Run “Table to Points” filter from the main menu Filters → Alphabetical;
- Determine the meaning of each column in the Properties panel and confirm it with the Apply button;
- Open new 3D View;
- Click the eye 👁 in the Pipeline Browser located left to the current TableToPoints;
- Set Point on the Representation list in the Properties panel;
- Click the Edit button in Colouring options and change the colour of path points;
- In the Pipeline Browser, change the "TableToPoints" name to "Path XX" (Fig. 5.9).



Properties (Path 00)

X Column x
Y Column y
Z Column z
☐ 2D Points
☐ Keep All Data Arrays

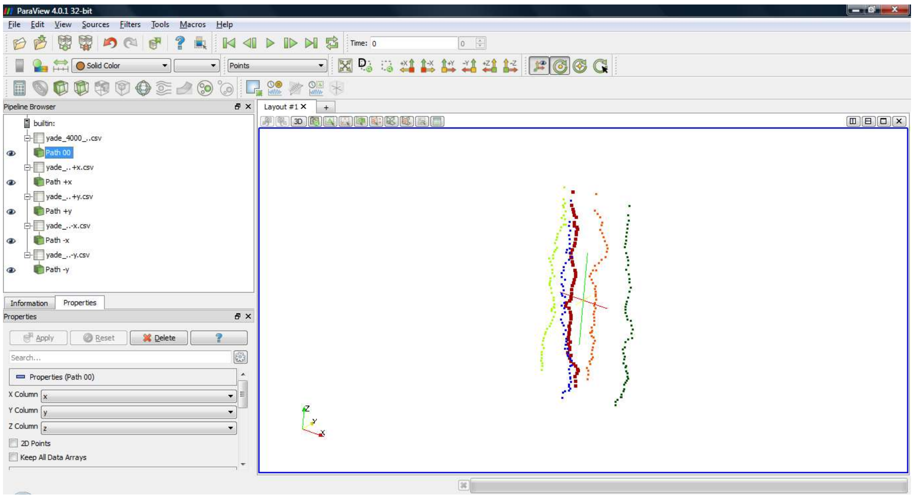
Display (GeometryRepresentation)

Representation Points

Fig. 5.9. Visualization of a path

Optionally, the size of points and the opacity may be changed in the Properties panel.

In the same way next paths may be added to the drawing (Fig. 5.10).



Properties (Path 00)

X Column x
Y Column y
Z Column z
☐ 2D Points
☐ Keep All Data Arrays

Fig. 5.10. Visualization of a few path inside the bed

5.5 The visualization of tetrahedrons surrounding the path using VTK files

This kind of visualization shows tetrahedrons by which goes the path.

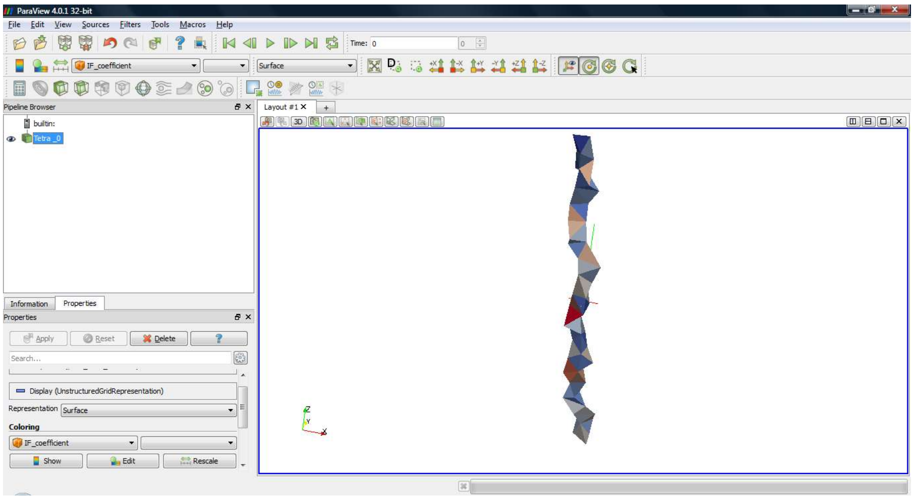
The visualization of the tetrahedrons may be done through following steps:

- Run the ParaView software;
- Open TheNameXX.res.vtk file and confirm it with the Apply button;
- In the Pipeline Browser change the "TheNameXX.res.vtk" name to "Tetra XX" (Fig. 5.11).

The triangles that form the tetrahedron basis may be coloured by a scalar (Fig. 5.12):

- Tetrahedron triangle basis area;
- Tetrahedron triangle basis perimeter;
- Correction coefficient for the Ideal Location in the searching method;

- The quality I_F coefficient;
- The angle between path sections;
- The angle between triangle normal and the Z-axis.

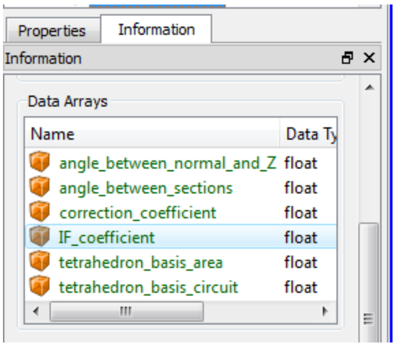


Display (UnstructuredGridRepresentation)

Representation: Surface

Coloring: IF_coefficient

Fig. 5.11. Visualization of tetrahedrons surrounding the path



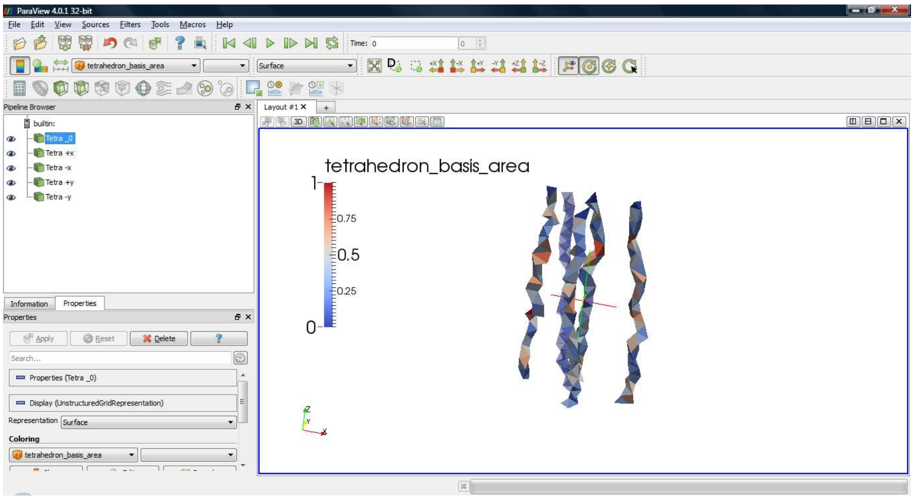
Data Arrays

Name	Data Ty
angle_between_normal_and_Z	float
angle_between_sections	float
correction_coefficient	float
IF_coefficient	float
tetrahedron_basis_area	float
tetrahedron_basis_circuit	float

Fig. 5.12. Available data arrays

The scalars give the same information as PLT scripts, but the way of visualization is different.

In the same way next tetrahedrons may be added to the drawing (Fig. 5.13).



Properties (Tetra_0)

Display (UnstructuredGridRepresentation)

Representation: Surface

Coloring: tetrahedron_basis_area

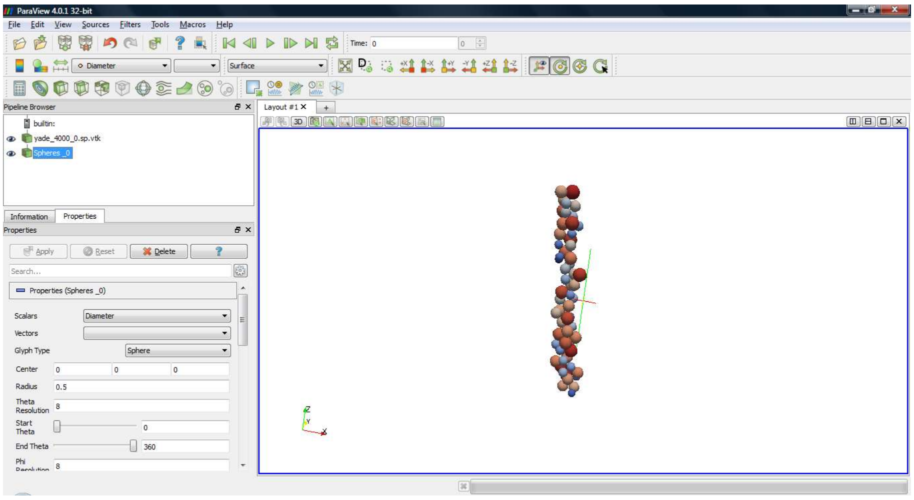
Fig. 5.13. Visualization of tetrahedrons surrounding the path, mapping by an output scalar

5.6 The visualization of spheres surrounding the path using VTK files

This kind of visualization shows spheres surrounding the path. Centres of these spheres form triangles and the tetrahedral structures, through which goes the path.

The visualization of the spheres may be done through following steps:

- Run the ParaView software;
- Open TheNameXX.sp.vtk file and confirm it with the Apply button;
- Turn the Glyph filter ;
- Set options for the Glyph filter:
  - Glyph type: Sphere;
  - Scale mode: Scalar;
  - Set scalar factor: 1;
- Press the Apply button;
- In the Pipeline Browser, change the "Glyph1" name to "Spheres XX" (Fig. 5.14).



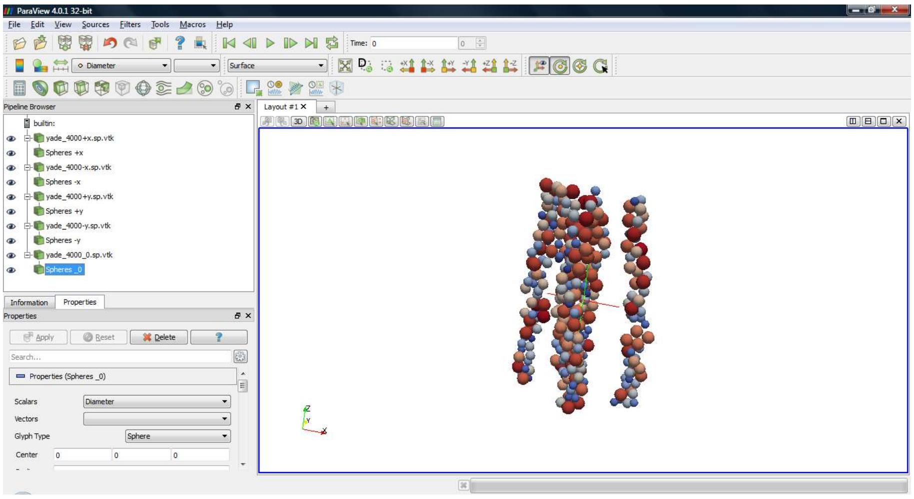
builtin:
yade_4000_0.sp.vtk
Spheres _0

Properties (Spheres _0)

Scalars: Diameter
Vectors:
Glyph Type: Sphere
Center: 0 0 0
Radius: 0.5
Theta Resolution: 8
Start Theta: 0
End Theta: 360
Phi Resolution: 8

Fig. 5.14. Visualization of spheres surrounding the path

In the same way next spheres may be added to the drawing (Fig. 5.15).



builtin:
yade_4000+x.sp.vtk
Spheres +x
yade_4000-x.sp.vtk
Spheres -x
yade_4000+y.sp.vtk
Spheres +y
yade_4000-y.sp.vtk
Spheres -y
yade_4000_0.sp.vtk
Spheres _0

Properties (Spheres _0)

Scalars: Diameter
Vectors:
Glyph Type: Sphere
Center: 0 0 0

Fig. 5.15. Visualization of spheres surrounding the path

5.7 Visualization of the tortuosity field

This kind of visualization shows a map of tortuosity values for the whole bed. It can be done by performing following steps:

- Run the ParaView software;
- Open Comp_tau_rg.vtk file in the Comparison subfolder and confirm it with the Apply button;
- Change the kind of Coloring to tortuosity.

Optional steps:

- Open a second window with the BarChartView kind;
- Select Histogram Filter from the main menu and confirm it with the Apply button.

Result of this visualisation is shown in Fig. 5.16.

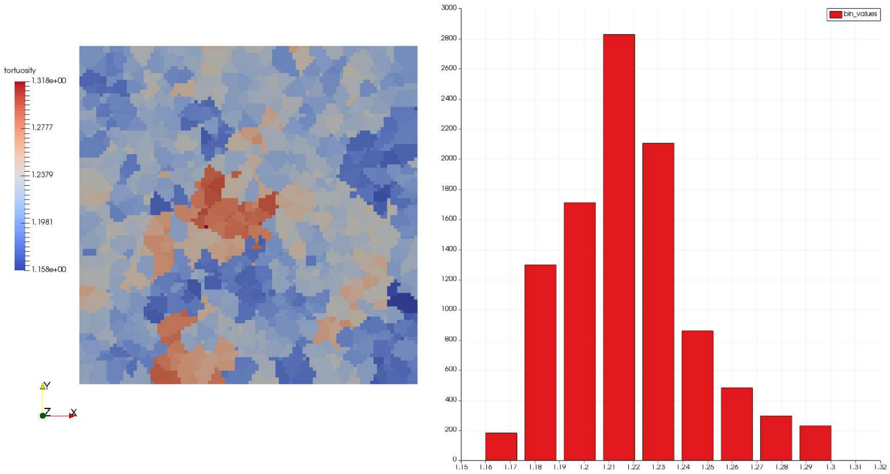


Fig. 5.16. Visualization of a tortuosity map

5.8 Mixing the kinds of visualization

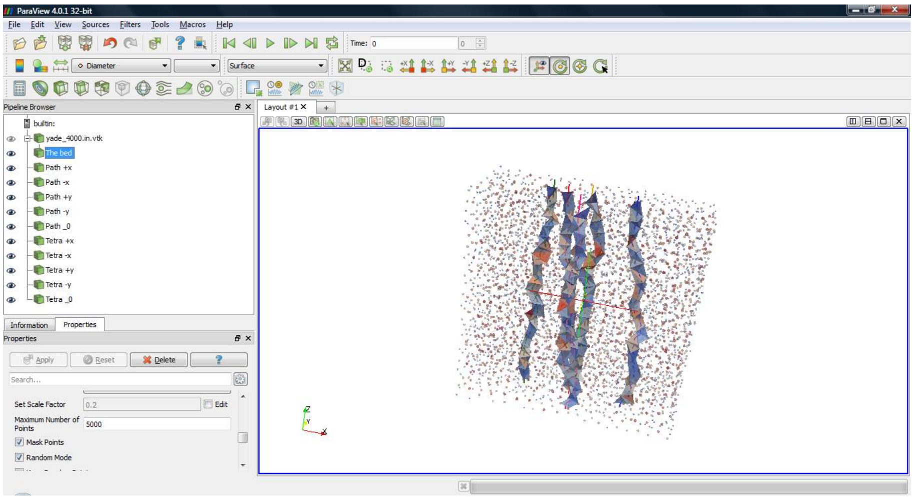
The ParaView software allows mixing different visualizations. To do this, one should add pieces, according to the chosen descriptions in the previous section.

In the Fig. 5.17 the bed is shown (Point 5.1), with five paths (Points 5.3) and five tetrahedrons sets for these paths (Point 5.5). The spheres are here rescaled and have smaller diameters. In this way, the inside of the bed is more visible.

In the Fig. 5.18 the bed is shown, (Point 5.1) with five paths (Points 5.3) and five particle sets for these paths (Point 5.6).

In the Fig. 5.19 a path is shown, (Points 5.3) with tetrahedral structures (Point 5.5) and spheres surrounding the path (Point 5.6).

The ParaView software allows using many visualization forms – for example the Clip filter (Fig. 5.20) – it is impossible to describe them all here. For more information, the Reader can refer to the ParaView tutorial.

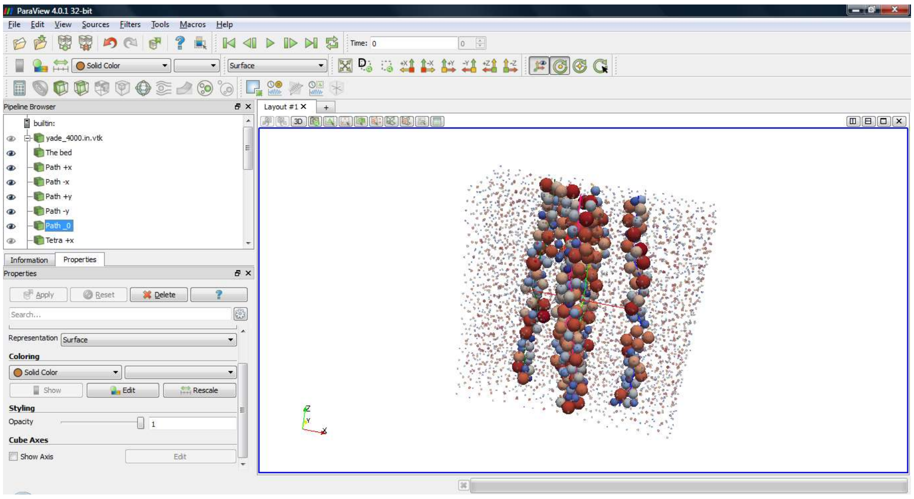


Diameter
Surface

builtin:
yade_4000.in.vtk
The bed
Path +x
Path -x
Path +y
Path -y
Path _0
Tetra +x
Tetra -x
Tetra +y
Tetra -y
Tetra _0

Set Scale Factor 0.2 □ Edit
Maximum Number of Points 5000
☑ Mask Points
☑ Random Mode

Fig. 5.17. An example of mixing visualization kinds

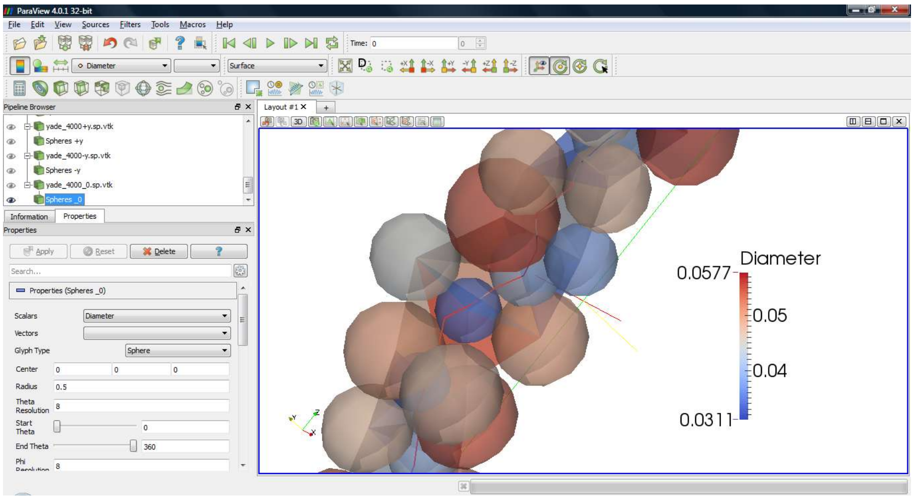


Solid Color
Surface

builtin:
yade_4000.in.vtk
The bed
Path +x
Path -x
Path +y
Path -y
Path _0
Tetra +x

Representation Surface
Coloring
Solid Color
Styling
Opacity 1
Cube Axes
□ Show Axis

Fig. 5.18. An example of mixing visualization kinds

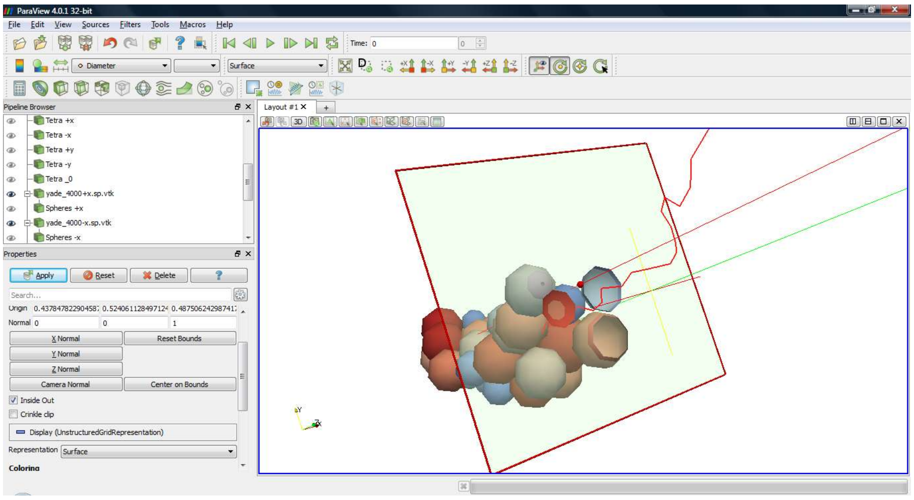


Diameter
Surface

yade_4000+y.sp.vtk
Spheres +y
yade_4000-y.sp.vtk
Spheres -y
yade_4000_0.sp.vtk
Spheres _0

Properties (Spheres _0)
Scalars Diameter
Vectors
Glyph Type Sphere
Center 0 0 0
Radius 0.5
Theta Resolution 8
Start Theta 0
End Theta 360
Phi Resolution 8

Fig. 5.19. An example of mixing visualization



- Tetra +x
- Tetra -x
- Tetra +y
- Tetra -y
- Tetra _0
- yade_4000+x.sp.vtk
- Spheres +x
- yade_4000-x.sp.vtk
- Spheres -x

Origin 0.437847822904587 0.524061128497124 0.487506242987417

Normal 0 0 1

☑ Inside Out

□ Crinkle clip

Display (UnstructuredGridRepresentation)

Representation Surface

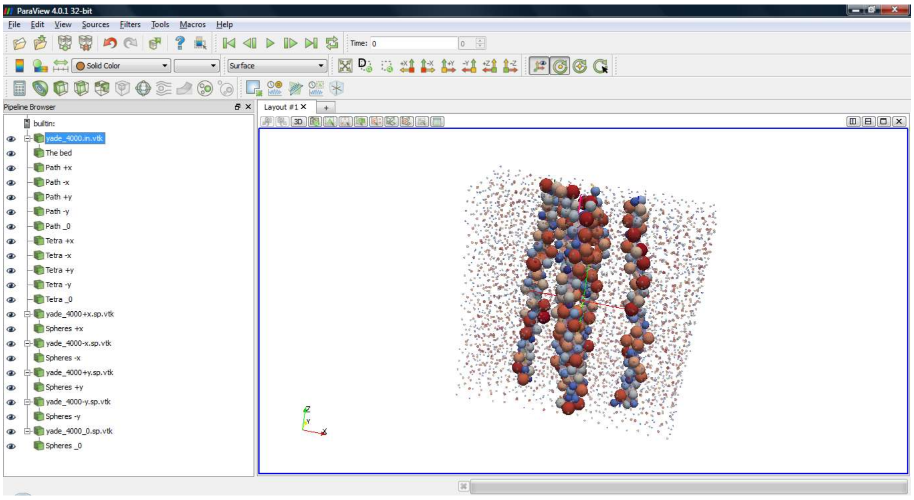
Fig. 5.20. An example of using the Clip filter

5.9 Saving states

Since the visualization takes a lot of time and contains many elements, the final visualization should be saved by "Save State" option in the File menu. Then ParaView running the saved scenes can be opened using "Load State" option. All elements may be shown or hidden using the Eye button 👁 on the left side of the element name.

In the PathFinder some examples of the *.pvsm files are included. After loading these files, a window with all used filenames is shown. The User must change these paths in accordance with the location of the PathFinder directory.

The file *.pvsm may be opened and all paths may be changed manually using "Find and Replace..." option in any text editor.



- builtin:
- yade_4000.in.vtk
- The bed
- Path +x
- Path -x
- Path +y
- Path -y
- Path _0
- Tetra +x
- Tetra -x
- Tetra +y
- Tetra -y
- Tetra _0
- yade_4000+x.sp.vtk
- Spheres +x
- yade_4000-x.sp.vtk
- Spheres -x
- yade_4000+y.sp.vtk
- Spheres +y
- yade_4000-y.sp.vtk
- Spheres -y
- yade_4000_0.sp.vtk
- Spheres _0

Fig. 5.21. A final visualization form for five paths with using VTK files

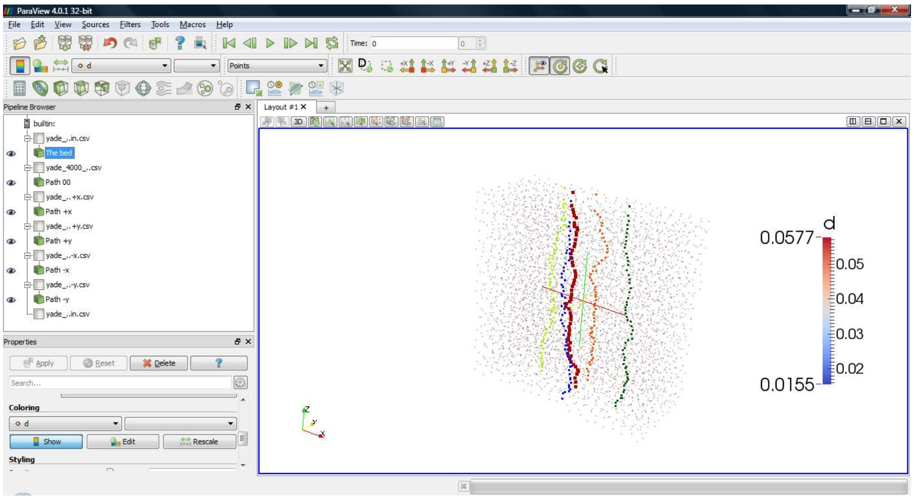
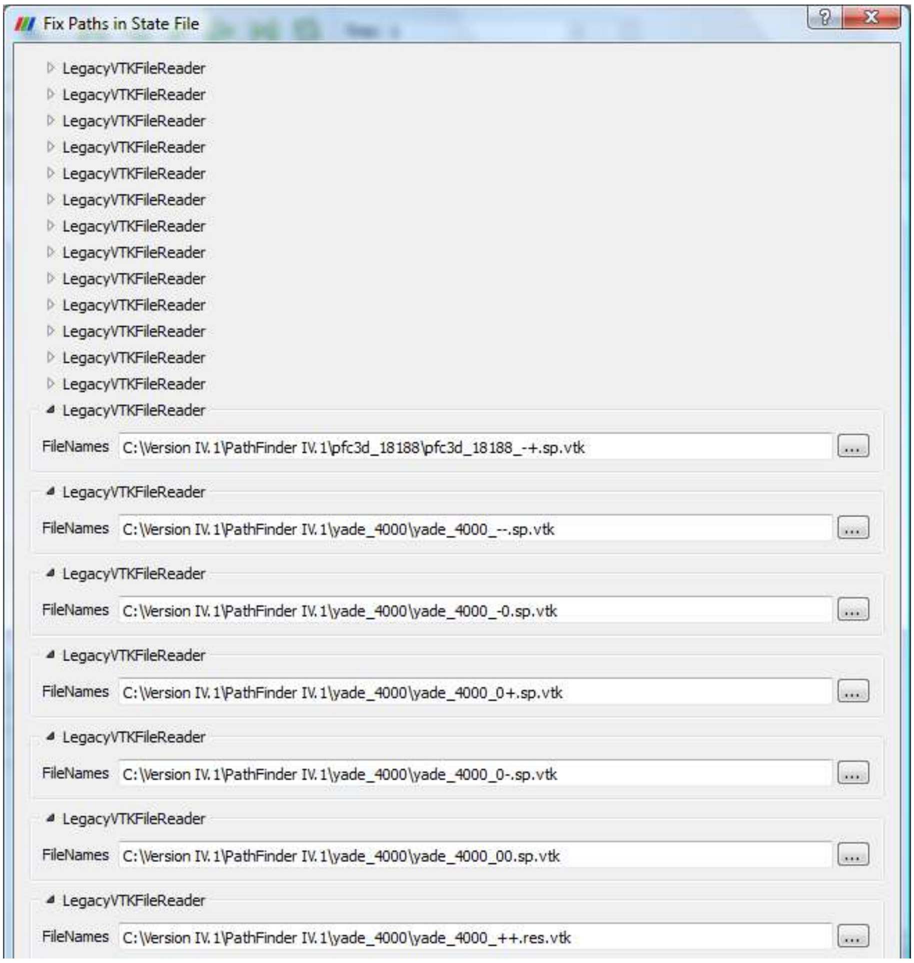


Fig. 5.22. A final visualization form for five paths with using CSV files


Fix Paths in State File

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

LegacyVTKFileReader

FileNames C:\Version IV.1\PathFinder IV.1\pfc3d_18188\pfc3d_18188_-+.sp.vtk

LegacyVTKFileReader

FileNames C:\Version IV.1\PathFinder IV.1\yade_4000\yade_4000_--.sp.vtk

LegacyVTKFileReader

FileNames C:\Version IV.1\PathFinder IV.1\yade_4000\yade_4000_-0.sp.vtk

LegacyVTKFileReader

FileNames C:\Version IV.1\PathFinder IV.1\yade_4000\yade_4000_0+.sp.vtk

LegacyVTKFileReader

FileNames C:\Version IV.1\PathFinder IV.1\yade_4000\yade_4000_0-.sp.vtk

LegacyVTKFileReader

FileNames C:\Version IV.1\PathFinder IV.1\yade_4000\yade_4000_00.sp.vtk

LegacyVTKFileReader

FileNames C:\Version IV.1\PathFinder IV.1\yade_4000\yade_4000_++.res.vtk

Fig. 5.23. A final visualization form for five paths with using CSV files

6 The visualisation in the MayaVi software

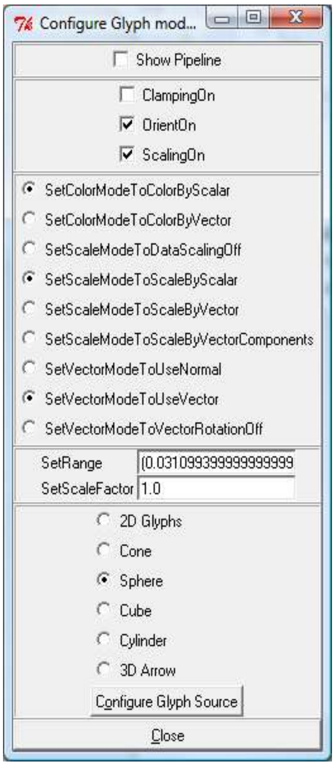
MayaVi[8] has two versions: the line to version 1.5 (no longer under development) and the line beginning with version 2.X. In this chapter, use of both versions is described.

6.1 The MayaVi 1.5

6.1.1 The visualization of the bed using VTK files

The visualization of the bed may be done in MayaVi 1.5 through following steps:

- Run the MayaVi software;
- Open TheName.in.vtk file;
- Choose the option "Modules – Glyph" from the "Vizualize" menu;
- Choose "Sphere" in the "Configure Glyph" window and turn off the "Clamping" option (Fig. 6.1); The result is shown in the Fig. 6.2.
- To obtain a more smoothed view choose "Configure Glyph Source" in the "Configure Glyph" window and increase the value of the parameter called SetThetaResolution (Fig. 6.3).
- Apply new settings (6.4).
- The spheres may be shown in a scale with the "SetScalarFactor" parameter in the "Configure Glyph Source" – a result with scalar factor equal to 0.1 is presented in the Fig. 6.5.
- To show the frames of the domain chose option "Modules – Outline" from the "Vizualize" menu.



□ Show Pipeline

□ ClampingOn
☑ OrientOn
☑ ScalingOn

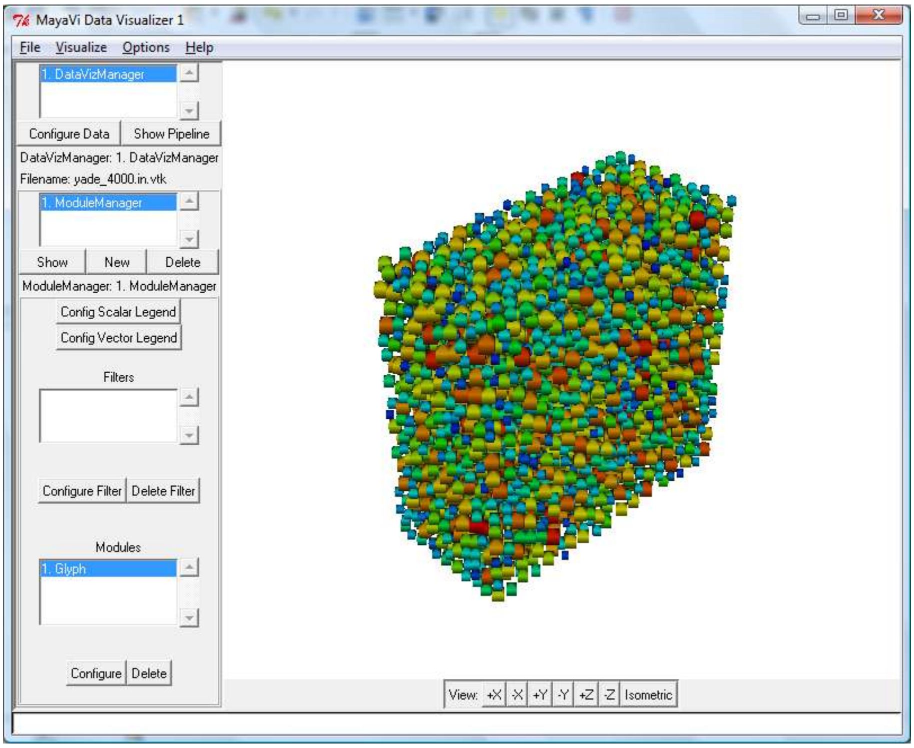
● SetColorModeToColorByScalar
○ SetColorModeToColorByVector
○ SetScaleModeToDataScalingOff
● SetScaleModeToScaleByScalar
○ SetScaleModeToScaleByVector
○ SetScaleModeToScaleByVectorComponents
○ SetVectorModeToUseNormal
● SetVectorModeToUseVector
○ SetVectorModeToVectorRotationOff

SetRange (0.0310993999999999
SetScaleFactor 1.0

○ 2D Glyphs
○ Cone
● Sphere
○ Cube
○ Cylinder
○ 3D Arrow

Fig. 6.1. Settings in the Configure Glyph window

[8] The MayaVi Data Visualizer [on-line]: URL: http://mayavi.sourceforge.net/.

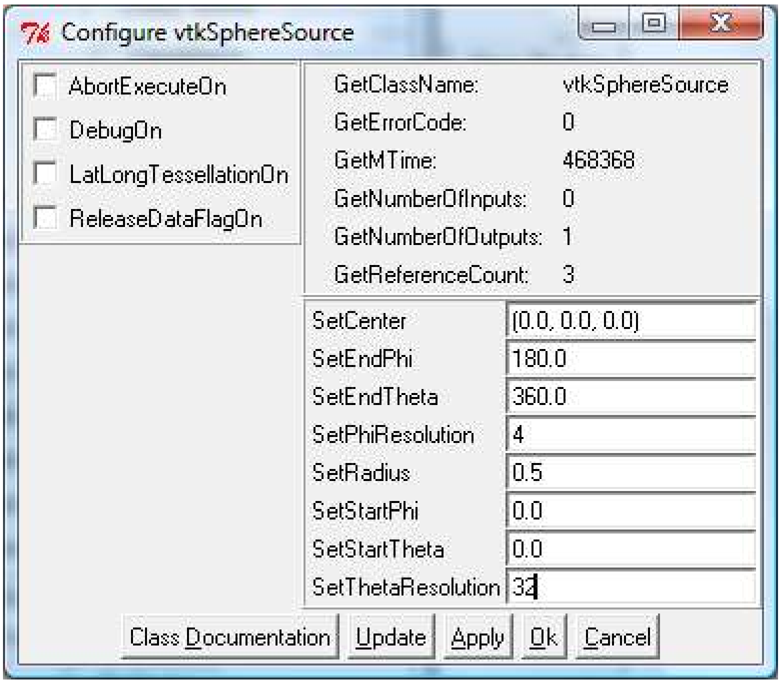


DataVizManager: 1. DataVizManager

Filename: yade_4000.in.vtk

ModuleManager: 1. ModuleManager

Fig. 6.2. Visualisation of the bed in MayaVi 1.5



□ AbortExecuteOn

□ DebugOn

□ LatLongTessellationOn

□ ReleaseDataFlagOn

GetClassName: vtkSphereSource

GetErrorCode: 0

GetMTime: 468368

GetNumberOfInputs: 0

GetNumberOfOutputs: 1

GetReferenceCount: 3

SetCenter (0.0, 0.0, 0.0)

SetEndPhi 180.0

SetEndTheta 360.0

SetPhiResolution 4

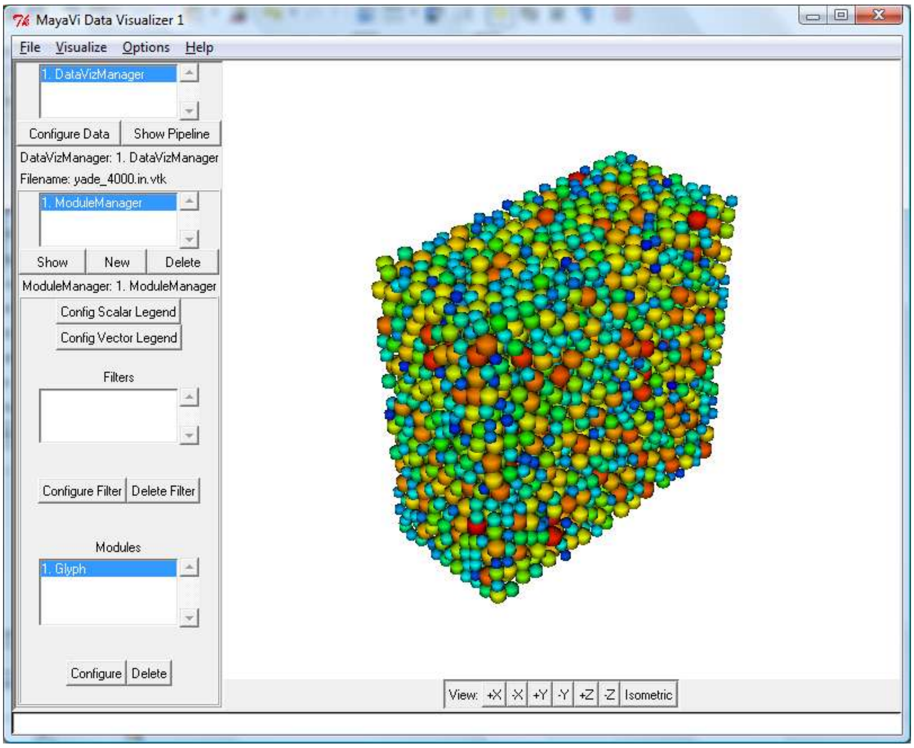
SetRadius 0.5

SetStartPhi 0.0

SetStartTheta 0.0

SetThetaResolution 32

Fig. 6.3. Configuring Sphere Source window

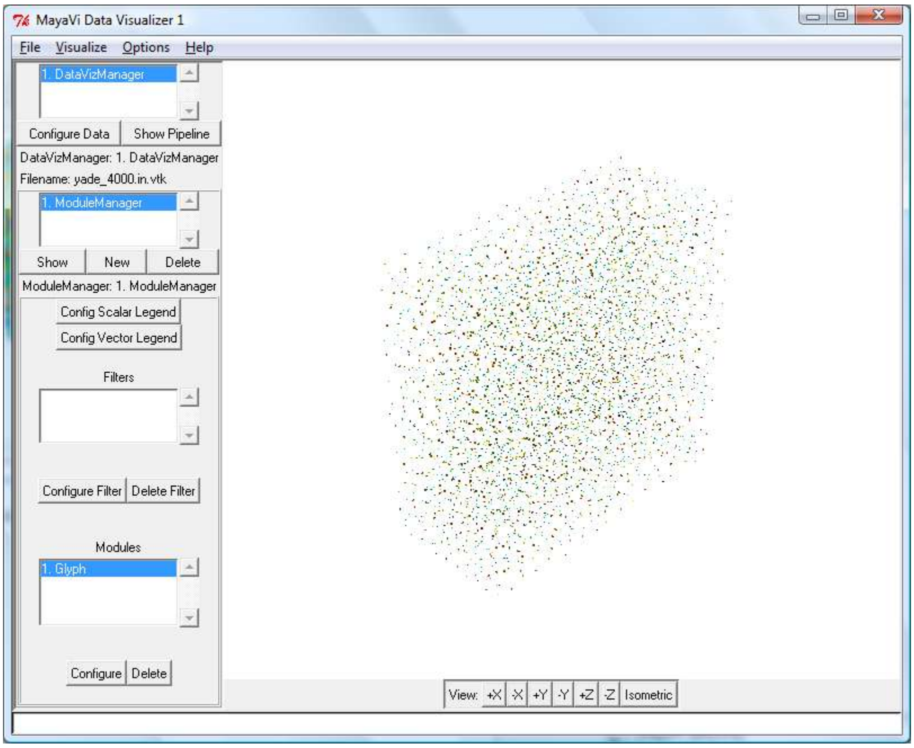


DataVizManager: 1. DataVizManager

Filename: yade_4000.in.vtk

ModuleManager: 1. ModuleManager

Fig. 6.4. Visualisation of the bed after smoothing



DataVizManager: 1. DataVizManager

Filename: yade_4000.in.vtk

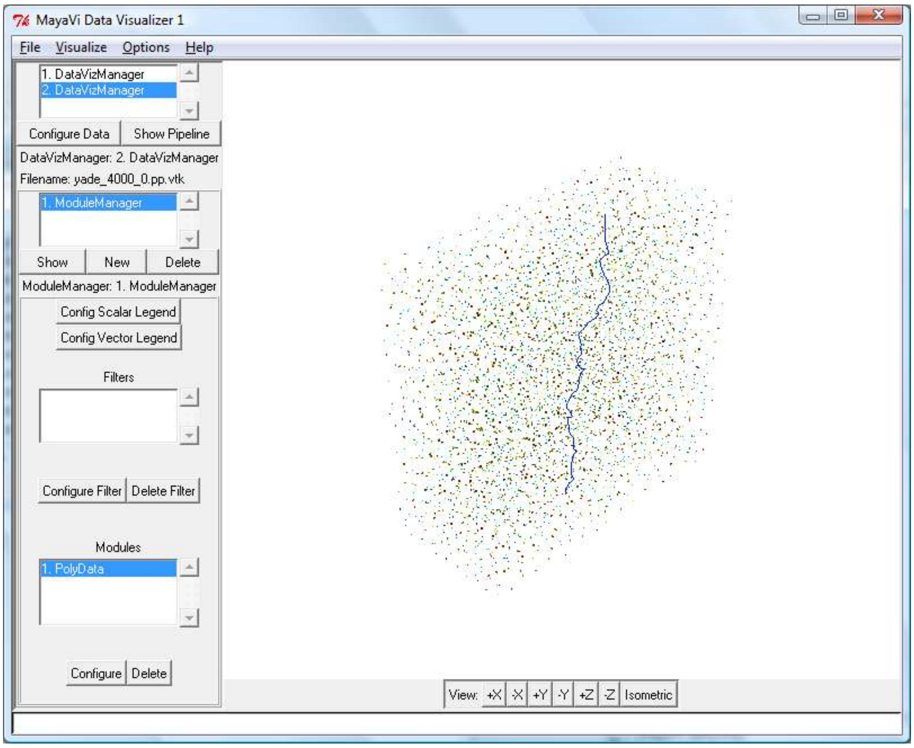
ModuleManager: 1. ModuleManager

Fig. 6.5. Visualisation of the bed after scaling

6.1.2 The visualization of the path using VTK files

The visualization of the path may be done through following steps:

- Open TheName_XX.pp.vtk file in MayaVi;
- Choose "Modules – PolyData" option from the "Vizualize" menu (Fig. 6.6).
- In the "Configure PolyData" window, set the line width, the line colour or the line transparency (Fig. 6.7).

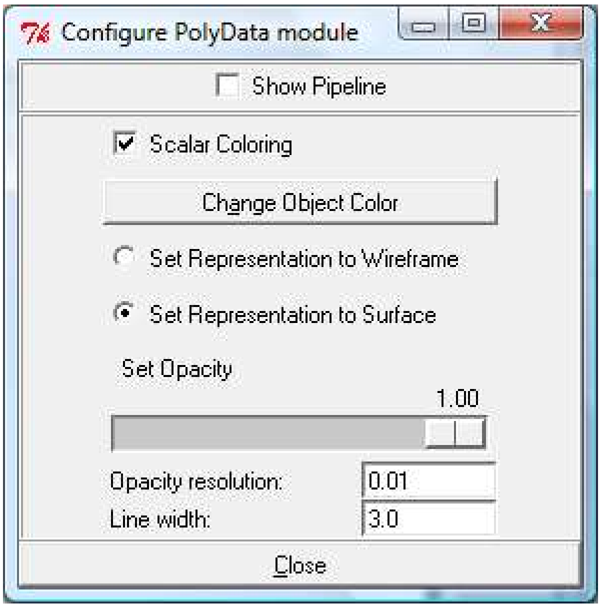


DataVizManager: 2. DataVizManager

Filename: yade_4000_0.pp.vtk

ModuleManager: 1. ModuleManager

Fig. 6.6. Visualisation of the path in the MayaVi 1.5



□ Show Pipeline

☑ Scalar Coloring

○ Set Representation to Wireframe

● Set Representation to Surface

Set Opacity 1.00

Opacity resolution: 0.01

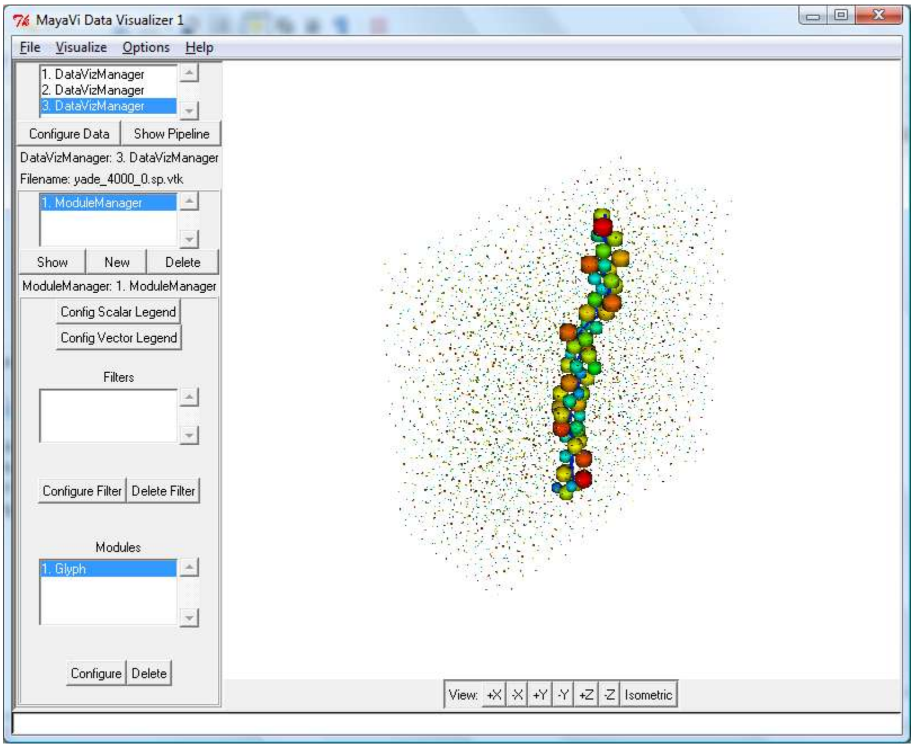
Line width: 3.0

Fig. 6.7. The Configure Poly Data module window

6.1.3 The visualization of spheres surrounding the path using VTK files

The visualization of the spheres may be done through following steps:

- Open TheNameXX.sp.vtk file;
- Choose "Modules – Glyph" from the "Vizualize" menu;
- Choose "Sphere" in the "Configure Glyph" window and turn off the "Clamping" option
- Choose "Configure Glyph Source" in the "Configure Glyph" window, increase the SetThetaResolution parameter and confirm it with the Apply button (Fig. 6.8).



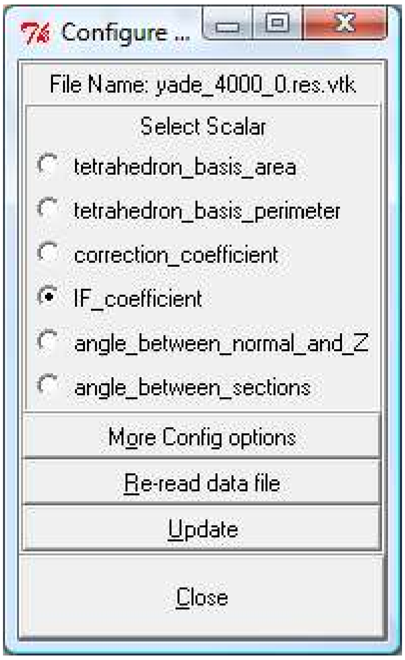
DataVizManager: 3. DataVizManager
Filename: yade_4000_0.sp.vtk
ModuleManager: 1. ModuleManager

Fig. 6.8. Visualization of spheres surrounding the path

6.1.4 The visualization of tetrahedrons surrounding the path using VTK files

The visualization of the tetrahedrons may be done through following steps:

- Open TheNameXX.res.vtk file;
- Choose the scalar for visualisation (Fig. 6.9);
- Choose "Modules – BandedSurfaceMap module" from the "Vizualize" menu (Fig. 6.10);
- To show a legend to the chosen scalar, turn on the "Show Legend" option in the "Configure LutHandler module" window (Fig. 6.11). The final results is shown in the Fig. 6.12.



Configure ...

File Name: yade_4000_0.res.vtk

Select Scalar

○ tetrahedron_basis_area

○ tetrahedron_basis_perimeter

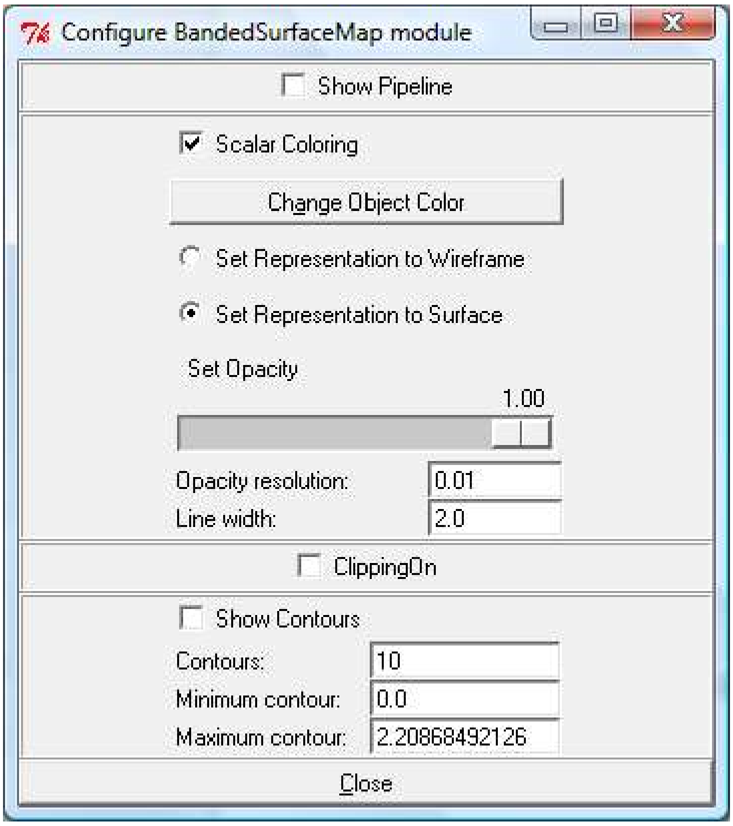
○ correction_coefficient

● IF_coefficient

○ angle_between_normal_and_Z

○ angle_between_sections

Fig. 6.9. The window with scalars available for colouring



Configure BandedSurfaceMap module

□ Show Pipeline

☑ Scalar Coloring

○ Set Representation to Wireframe

● Set Representation to Surface

Set Opacity 1.00

Opacity resolution: 0.01

Line width: 2.0

□ ClippingOn

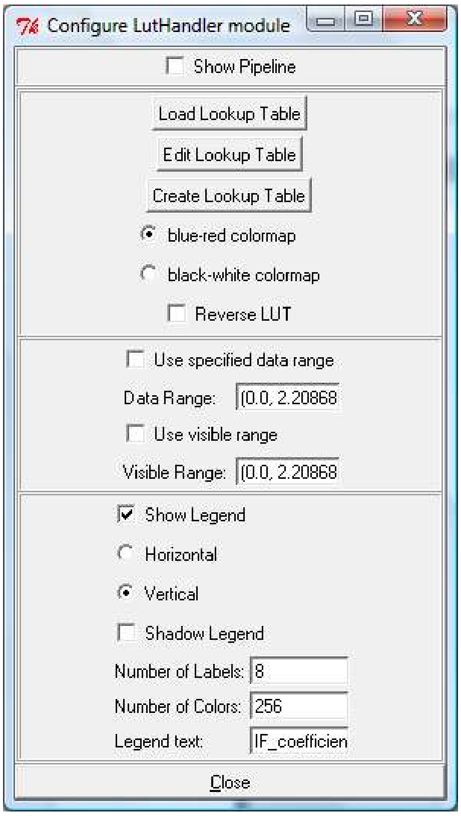
□ Show Contours

Contours: 10

Minimum contour: 0.0

Maximum contour: 2.20868492126

Fig. 6.10. The window with other options of scalar visualisation



□ Show Pipeline

● blue-red colormap

○ black-white colormap

□ Reverse LUT

□ Use specified data range

Data Range: (0.0, 2.20868

□ Use visible range

Visible Range: (0.0, 2.20868

☑ Show Legend

○ Horizontal

● Vertical

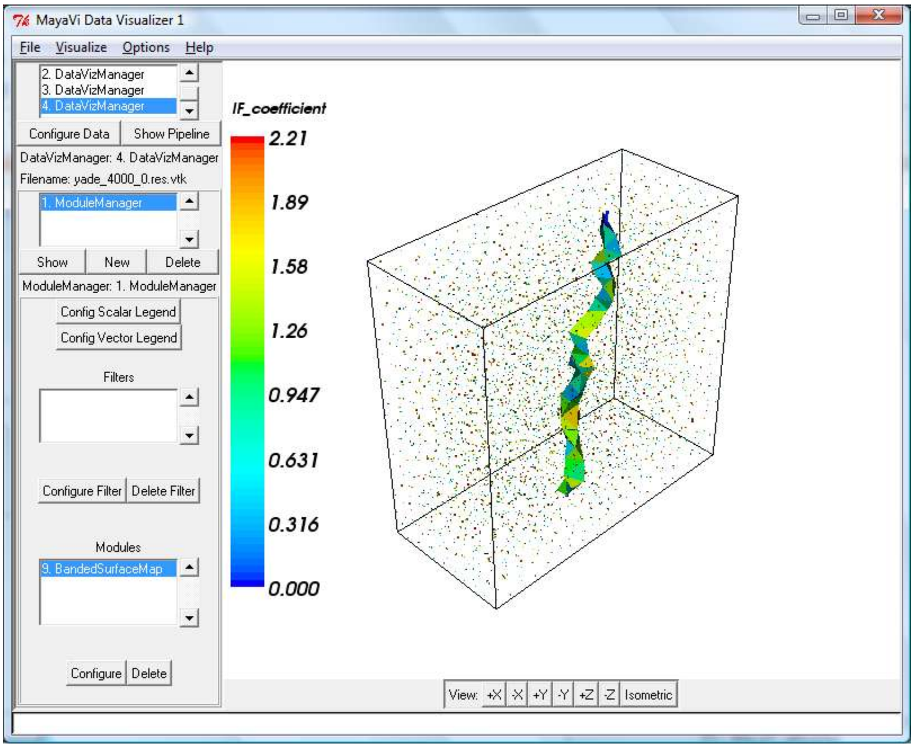
□ Shadow Legend

Number of Labels: 8

Number of Colors: 256

Legend text: IF_coefficien

Fig. 6.11. The window with options of scalar legend visualisation



DataVizManager: 4. DataVizManager

Filename: yade_4000_0.res.vtk

ModuleManager: 1. ModuleManager

Fig. 6.12. Visualization of tetrahedrons surrounding the path

6.2 The MayaVi 2.X

6.2.1 Visualization of the bed

The visualization of the bed may be done in MayaVi 2.X through following steps:

1. Run the MayaVi2 application – in Unix/Linux OS type in the console:

```
mayavi2
```

2. In the "Engine Tree View" click the position "Add Data Source" (Fig. 6.13).

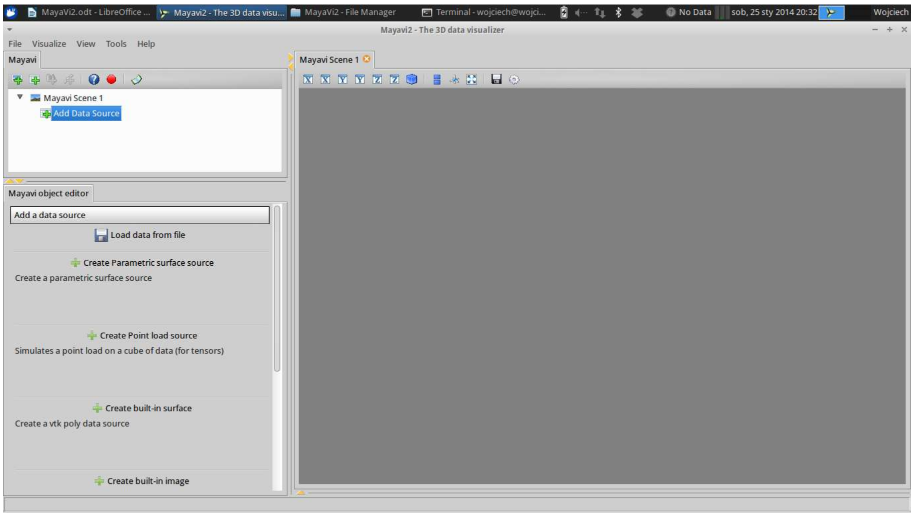

MayaVi2.odt - LibreOffice ...
Mayavi2 - The 3D data visu...
MayaVi2 - File Manager
Terminal - wojciech@wojci...
No Data
sob, 25 sty 2014 20:32
Wojciech
Mayavi2 - The 3D data visualizer
File
Visualize
View
Tools
Help
Mayavi
Mayavi Scene 1
Add Data Source
Mayavi object editor
Add a data source
Load data from file
Create Parametric surface source
Create a parametric surface source
Create Point load source
Simulates a point load on a cube of data (for tensors)
Create built-in surface
Create a vtk poly data source
Create built-in image

Fig. 6.13. The "Engine Tree View" in MayaVi 2.X

3. In the "Object Editor" click the "Load data from file" button and choose the TheName.in.vtk file (Fig. 6.14). The same effect may be obtain by using the button ⊞ on the window header.

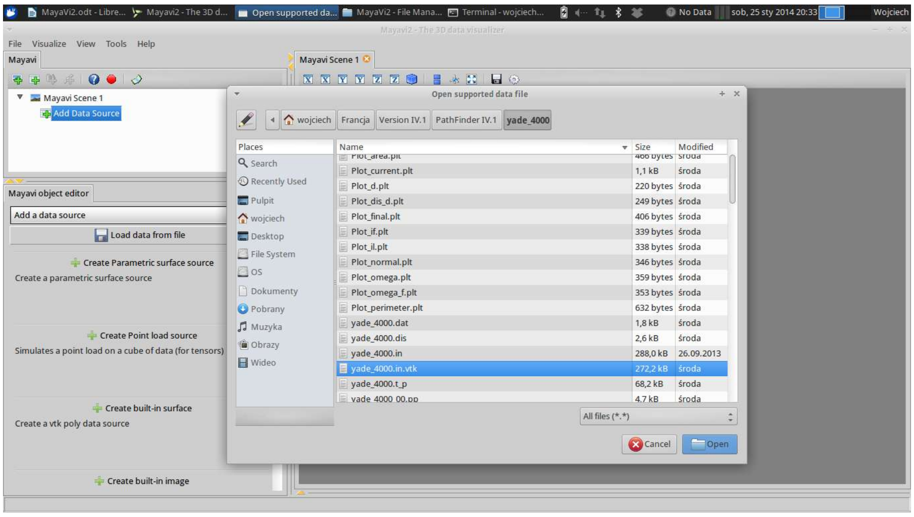

MayaVi2.odt - Libre...
Mayavi2 - The 3D d...
Open supported da...
MayaVi2 - File Mana...
Terminal - wojciech...
No Data
sob, 25 sty 2014 20:33
Wojciech
Mayavi2 - The 3D data visualizer
File
Visualize
View
Tools
Help
Mayavi
Mayavi Scene 1
Add Data Source
Mayavi object editor
Add a data source
Load data from file
Create Parametric surface source
Create a parametric surface source
Create Point load source
Simulates a point load on a cube of data (for tensors)
Create built-in surface
Create a vtk poly data source
Create built-in image
Open supported data file
wojciech
Francja
Version IV.1
PathFinder IV.1
yade_4000
Places
Search
Recently Used
Pulpit
wojciech
Desktop
File System
OS
Dokumenty
Pobrany
Muzyka
Obrazy
Wideo
Name
Size
Modified
Plot_area.plt 400 bytes środa
Plot_current.plt 1,1 kB środa
Plot_d.plt 220 bytes środa
Plot_dis_d.plt 249 bytes środa
Plot_final.plt 406 bytes środa
Plot_if.plt 339 bytes środa
Plot_il.plt 338 bytes środa
Plot_normal.plt 346 bytes środa
Plot_omega.plt 359 bytes środa
Plot_omega_f.plt 353 bytes środa
Plot_perimeter.plt 632 bytes środa
yade_4000.dat 1,8 kB środa
yade_4000.dis 2,6 kB środa
yade_4000.in 288,0 kB 26.09.2013
yade_4000.in.vtk 272,2 kB środa
yade_4000.t_p 68,2 kB środa
yade_4000_00.pp 4,7 kB środa
All files (*.*)
Cancel
Open

Fig. 6.14. Loading a new data source from a file

4. In the "Engine Tree View" click the"Add module or filter" position (Fig. 6.15).

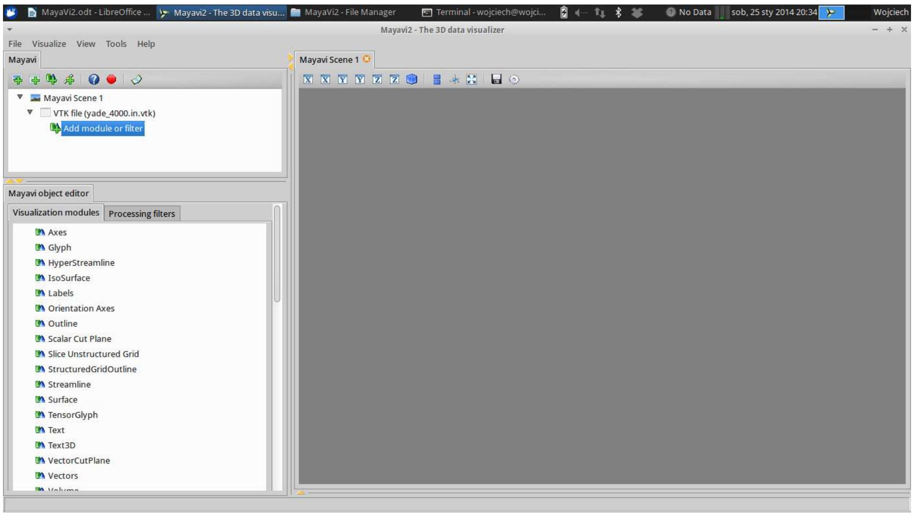
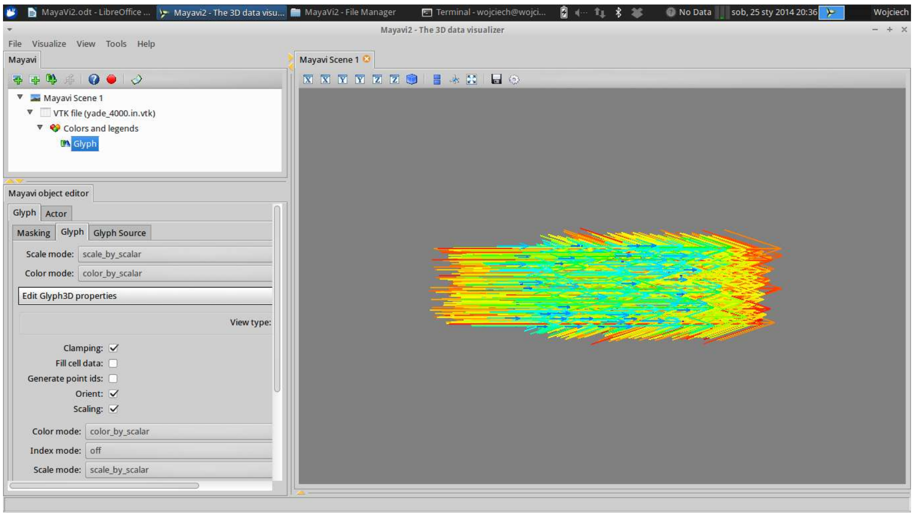


Fig. 6.15. The list of modules for visualization

5. Select the “Glyph” option in the “Visualization modules” tab and confirm it with the “Glyph” button or with double mouse click (Fig. 6.16).



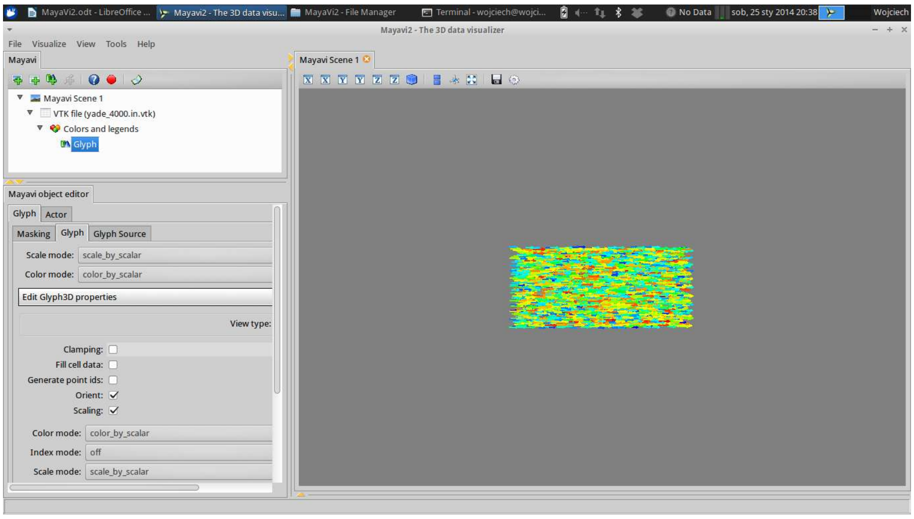
Scale mode: scale_by_scalar
Color mode: color_by_scalar

Clamping: ☑
Fill cell data: ☐
Generate point ids: ☐
Orient: ☑
Scaling: ☑

Color mode: color_by_scalar
Index mode: off
Scale mode: scale_by_scalar

Fig. 6.16. Default visualization of glyphs in MayaVi 2.X

6. Turn off the “Clamping” option in the “Glyph” tab (Fig. 6.17).
7. In the “Engine Tree View” click the “Glyph” position and on the “Glyph Source” tab select the “SphereSource” position on the selection list.
8. The quality of showed spheres may be corrected through increasing the values of “Phi resolution” or “Theta resolution” options.
9. To set the scene orientation and scale use the “Isometric View” button 🧊 in the header of the scene window (Fig. 6.18).
10. The transparency may be set on the “Actor” tab by using the “Opacity” option (Fig. 6.19).
11. To scale the particle diameters use the “Scale factor” on the “Glyph” tab (Fig. 6.20).
12. To show the scalar legend (particle diameters) click the “Colors and legend” option on the “Engine Tree View” and turn of the “Show legend” option (Fig. 6.21).
13. To set the white background of the scene click on the “Mayavi Scene 1” option and change values in the “background” edit on the “Scene” tab. In this case the description of the legend will be not visible – the “Foreground” value must be changed to black (Fig. 6.21).



- Mayavi Scene 1
  - VTK file (yade_4000.in.vtk)
    - Colors and legends
      - Glyph

Scale mode: scale_by_scalar

Color mode: color_by_scalar

Clamping: □

Fill cell data: □

Generate point ids: □

Orient: ✓

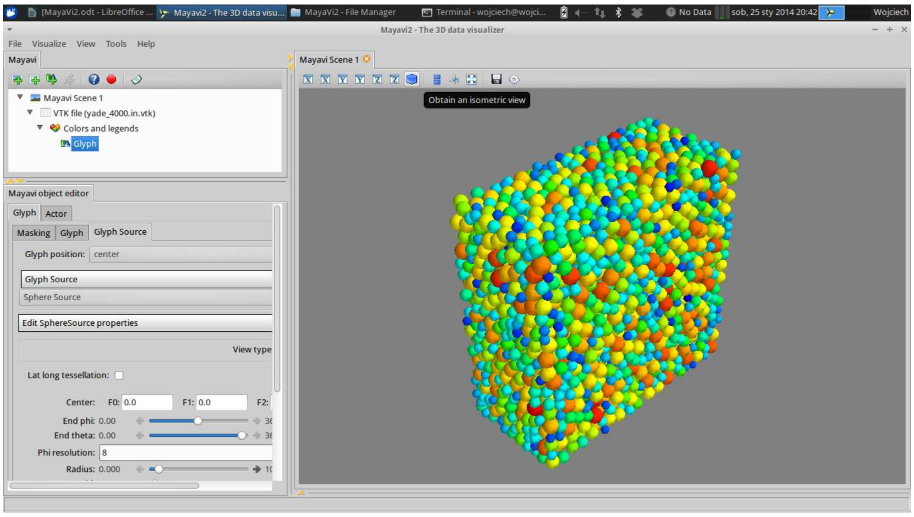
Scaling: ✓

Color mode: color_by_scalar

Index mode: off

Scale mode: scale_by_scalar

Fig. 6.17. Visualization of the bed after turning off of the clamping



- Mayavi Scene 1
  - VTK file (yade_4000.in.vtk)
    - Colors and legends
      - Glyph

Glyph position: center

Lat long tessellation: □

Center: F0: 0.0 F1: 0.0 F2:

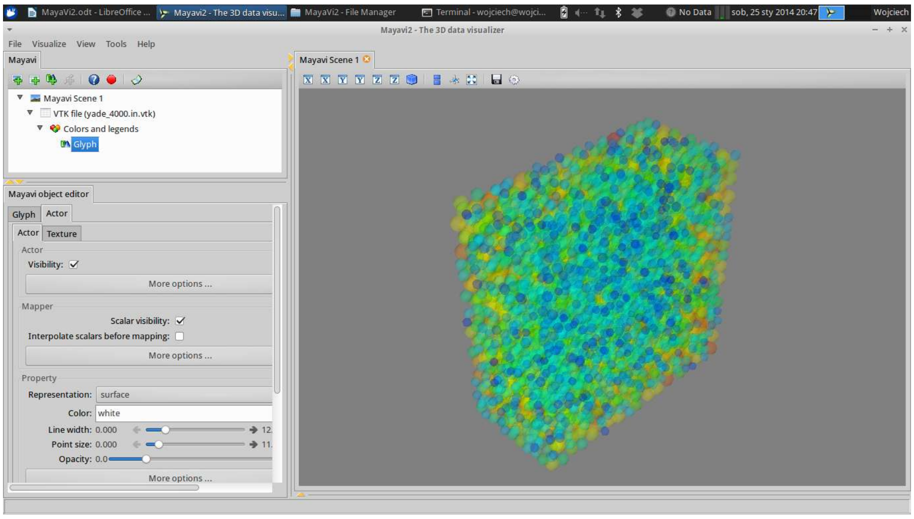
End phi: 0.00

End theta: 0.00

Phi resolution: 8

Radius: 0.000

Fig. 6.18. Visualization of the bed with the use of the sphere glyphs



- Mayavi Scene 1
  - VTK file (yade_4000.in.vtk)
    - Colors and legends
      - Glyph

Actor

Visibility: ✓

Mapper

Scalar visibility: ✓

Interpolate scalars before mapping: □

Property

Representation: surface

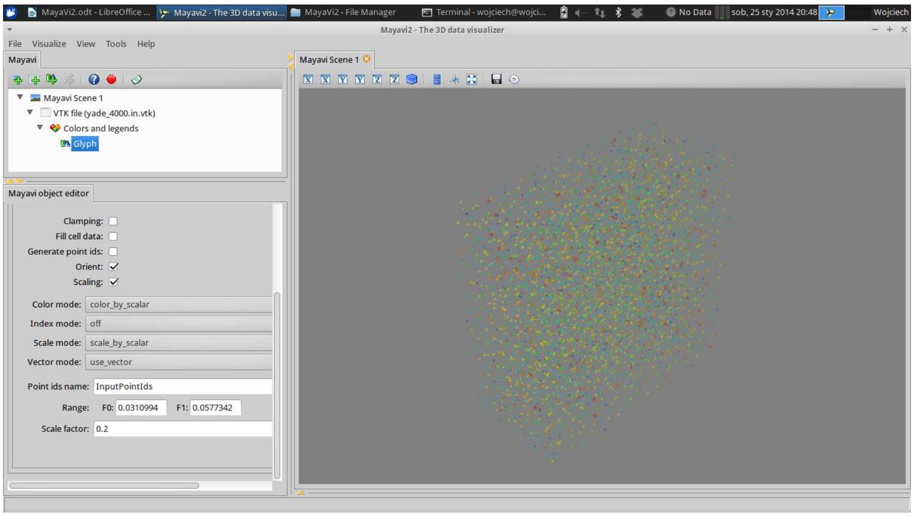
Color: white

Line width: 0.000

Point size: 0.000

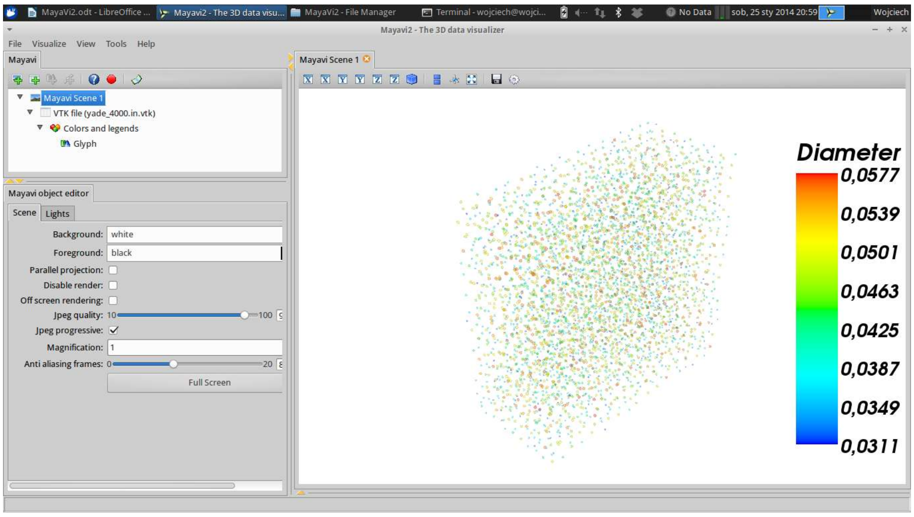
Opacity: 0.0

Fig. 6.19. Visualization of the bed with opacity equal to 0.2



Clamping: □
Fill cell data: □
Generate point ids: □
Orient: ✓
Scaling: ✓
Color mode: color_by_scalar
Index mode: off
Scale mode: scale_by_scalar
Vector mode: use_vector
Point ids name: InputPointIds
Range: F0: 0.0310994 F1: 0.0577342
Scale factor: 0.2

Fig. 6.20. Visualization of the bed with the scale factor equal to 0.2



Background: white
Foreground: black
Parallel projection: □
Disable render: □
Off screen rendering: □
Jpeg quality: 10 – 100
Jpeg progressive: ✓
Magnification: 1
Anti aliasing frames: 0 – 20
Full Screen

Fig. 6.21. Displaying the legend of the particle diameter scalar

6.2.2 Visualization of the path

The visualization of the bed may be done in MayaVi 2.X through following steps:

1. Click the "Add a data source" button on the header in the "Engine Tree View" window and open the TheName.pp.vtk file.
2. In the "Engine Tree View" click the "Add module or filter" position (Fig. 6.22).
3. Select the "Glyph" option in the "Visualization modules" tab and confirm it with the "Glyph" button or with double mouse click.
4. Turn off the "Clamping" option in the "Glyph" tab.
5. In the "Engine Tree View" click the "Glyph" position and on the "Glyph Source" tab select the "SphereSource" position on the selection list (Fig. 6.23).
6. Decrease the value of the "Scale Factor" on the "Glyph" tab (Fig. 6.24).
7. Change the "Color" on the "Actor" tab (Fig. 6.25).

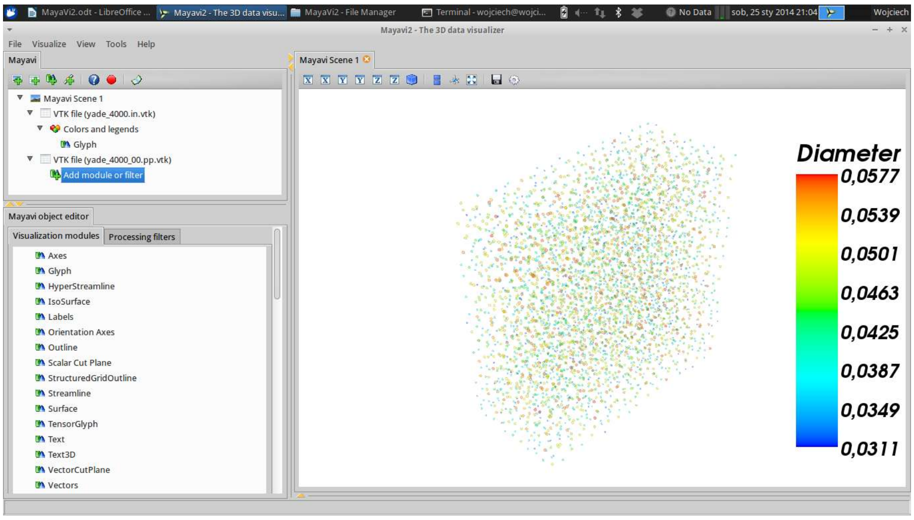
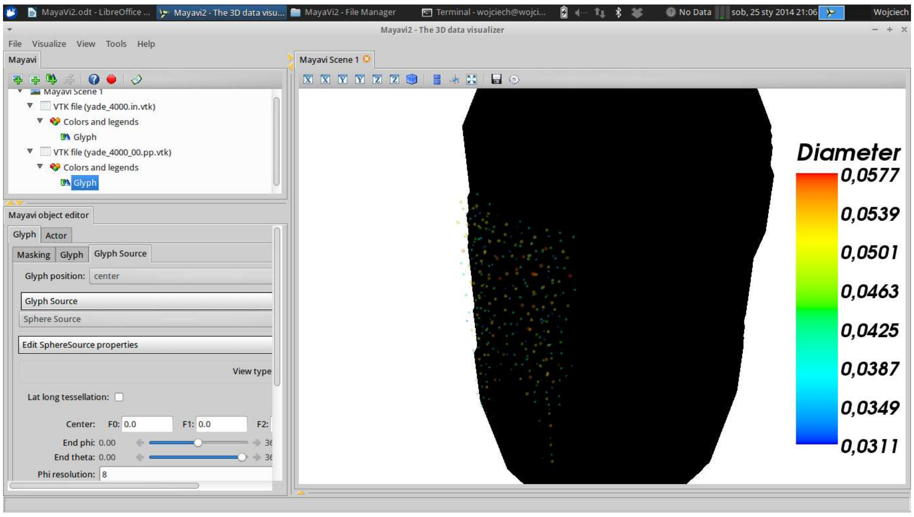


Fig. 6.22. Adding a new VTK file



Glyph position: center

Glyph Source: Sphere Source

Lat long tessellation: □

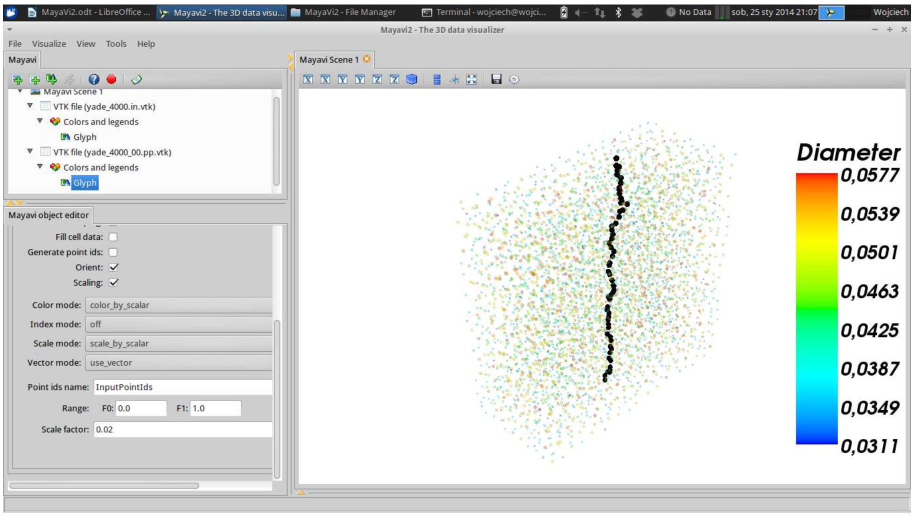
Center: F0: 0.0 F1: 0.0 F2:

End phi: 0.00

End theta: 0.00

Phi resolution: 8

Fig. 6.23. Default visualization of path points



Fill cell data: □

Generate point ids: □

Orient: ☑

Scaling: ☑

Color mode: color_by_scalar

Index mode: off

Scale mode: scale_by_scalar

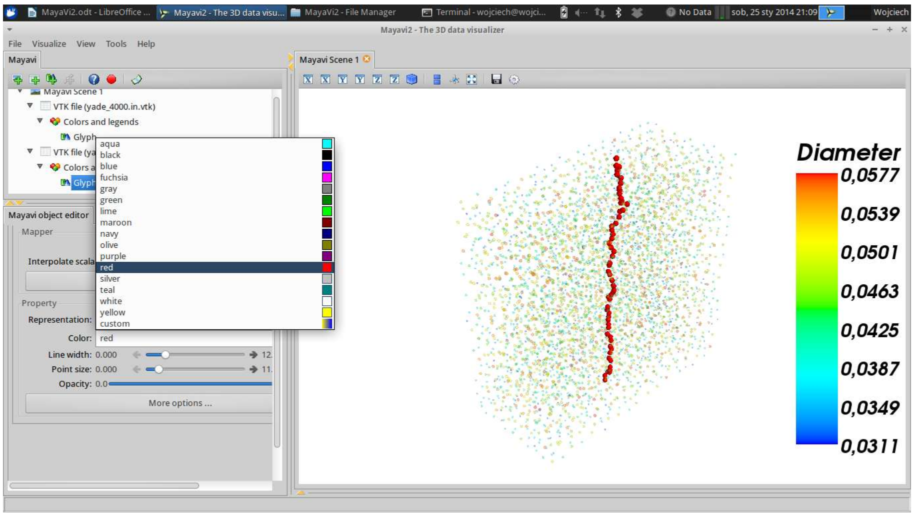
Vector mode: use_vector

Point ids name: InputPointIds

Range: F0: 0.0 F1: 1.0

Scale factor: 0.02

Fig. 6.24. Visualization of path points with scale factor equal to 0.02



Color: red

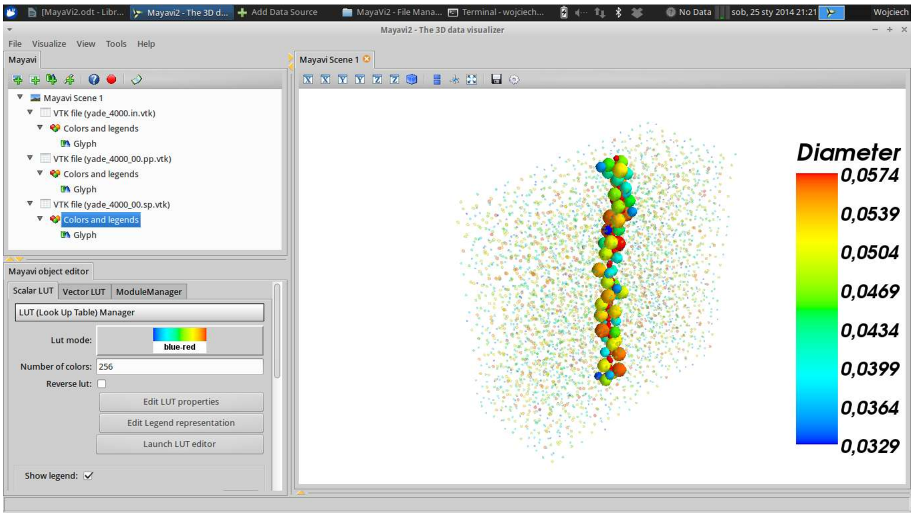
Fig. 6.25. Visualization of path points using a colour

6.2.3 Visualization of the spheres

The visualization of the bed may be done in MayaVi 2.X through following steps:

1. Click the “Add a data source” button on the header in the “Engine Tree View” window and open the TheName.sp.vtk file.
2. In the “Engine Tree View” click the “Add module or filter” position .
3. Select the “Glyph” option in the “Visualization modules” tab and confirm it with the “Glyph” button or with double mouse click.
4. Turn off the “Clamping” option in the “Glyph” tab.
5. In the “Engine Tree View” click the “Glyph” position and on the “Glyph Source” tab select the “SphereSource” position on the list.
6. To show the scalar legend (particle diameters) click the “Colors and legend” option on the “Engine Tree View” and turn of the “Show legend” option (remember to unselect this options in settings of other Glyph visualization).

The final visualization is shown in the Fig. 6.26.



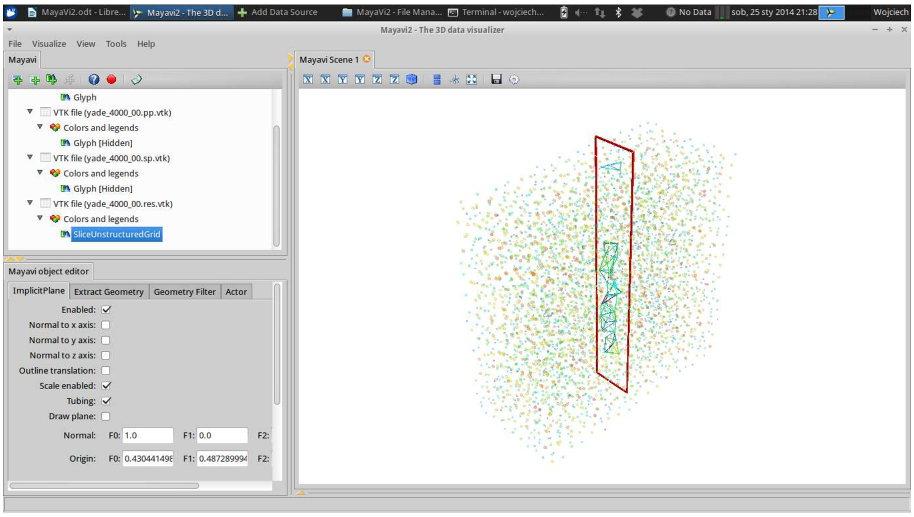
Lut mode: blue-red
Number of colors: 256
Reverse lut: □
Show legend: ☑

Fig. 6.26. Visualization of spheres surrounding a path

6.2.4 Visualization of the tetrahedrons

The visualization of the bed may be done in MayaVi 2.X through following steps:

1. Click the “Add a data source” button on the header in the “Engine Tree View” window and open the TheName.res.vtk file.
2. In the “Engine Tree View” click the “Add module or filter” position.
3. Select the “SliceUnstructuredGrid” option in the “Visualization modules” tab and confirm it with the “SliceUnstructuredGrid” button or with double mouse click (unselect all legends and hide other visualizations if needed) (Fig. 6.27).



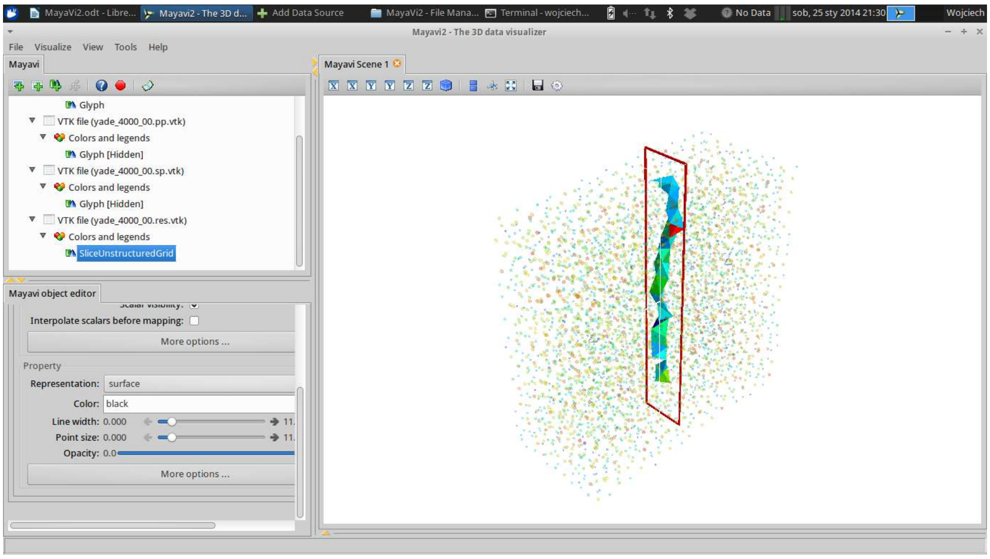
Glyph
VTK file (yade_4000_00.pp.vtk)
Colors and legends
Glyph [Hidden]
VTK file (yade_4000_00.sp.vtk)
Colors and legends
Glyph [Hidden]
VTK file (yade_4000_00.res.vtk)
Colors and legends
SliceUnstructuredGrid

Enabled: ☑
Normal to x axis: ☐
Normal to y axis: ☐
Normal to z axis: ☐
Outline translation: ☐
Scale enabled: ☑
Tubing: ☑
Draw plane: ☐
Normal: F0: 1.0 F1: 0.0 F2:
Origin: F0: 0.430441498 F1: 0.487289994 F2:

Fig. 6.27. Default visualization of an unstructured grid with a slice

The visibility of a visualization may be turned off/on in two ways: by using the “Visibility” option in the “Actor” tab or by using the RBM over the “Glyph” option in the “Engine Tree View”.

4. Unselect the “Extract only boundary cells” option on the “Extract geometry” tab.
5. Set the “Representation” on the “Actor” tab as “surface” (Fig. 6.28).

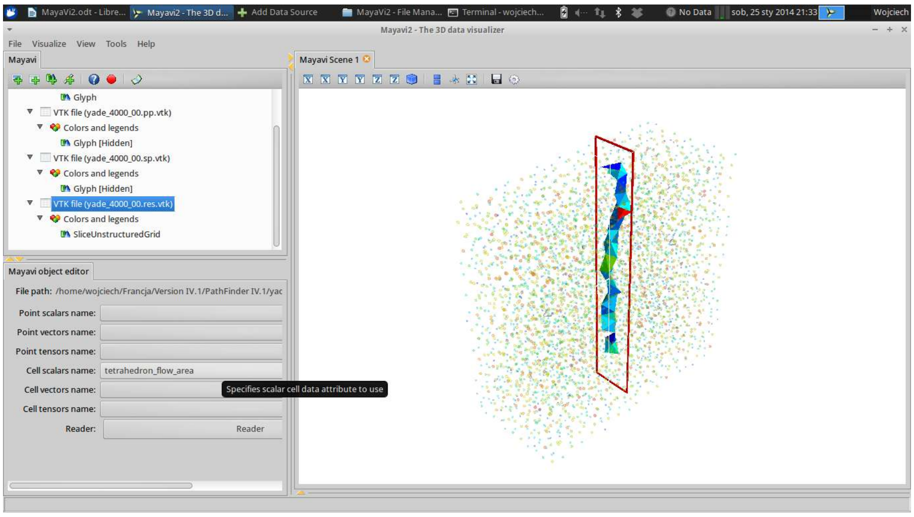


Glyph
VTK file (yade_4000_00.pp.vtk)
Colors and legends
Glyph [Hidden]
VTK file (yade_4000_00.sp.vtk)
Colors and legends
Glyph [Hidden]
VTK file (yade_4000_00.res.vtk)
Colors and legends
SliceUnstructuredGrid

Scalar visibility: ☑
Interpolate scalars before mapping: ☐
Representation: surface
Color: black
Line width: 0.000
Point size: 0.000
Opacity: 0.0

Fig. 6.28. Visualization of tetrahedrons forming a path

6. To change the scalar used for visualization click on the name of TheName.res.vtk file an set the “Cell scalars name”.


Mayavi2 - The 3D data visualizer
File Visualize View Tools Help
Mayavi
Glyph
VTK file (yade_4000_00.pp.vtk)
Colors and legends
Glyph [Hidden]
VTK file (yade_4000_00.sp.vtk)
Colors and legends
Glyph [Hidden]
VTK file (yade_4000_00.res.vtk)
Colors and legends
SliceUnstructuredGrid
Mayavi Scene 1
Mayavi object editor

File path: /home/wojciech/Francja/Version IV.1/PathFinder IV.1/yac

Point scalars name:

Point vectors name:

Point tensors name:

Cell scalars name: tetrahedron_flow_area

Cell vectors name:

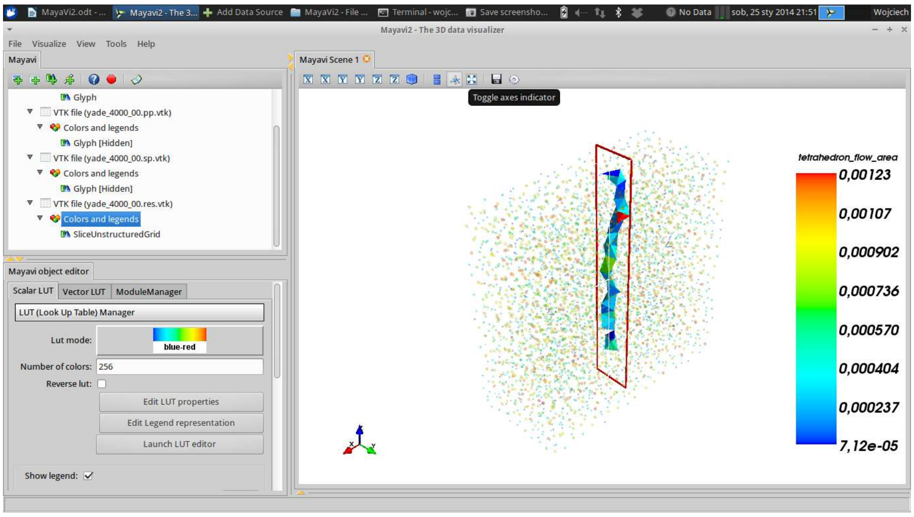
Cell tensors name:

Reader: Reader

Specifies scalar cell data attribute to use

Fig. 6.29. Changing the scalar for colouring the tetrahedrons

7. To show axis of the current coordinate system click the button “Toggle axes indicator” on the header of the scene window.
8. To show the scalar legend (for a scalar set like above) click the “Colours and legend” option on the “Engine Tree View” and turn of the “Show legend” option.



Lut mode: blue-red

Number of colors: 256

Reverse lut: □

Show legend: ☑

Fig. 6.30. A final visualization of tetrahedrons forming a path

9. To save the visualization in a popular file format use the “Save scene as” option in the “File” menu (Fig. 6.31).
10. To save the scene use the “Save visualization” option in the “File” main menu (Fig. 6.32).

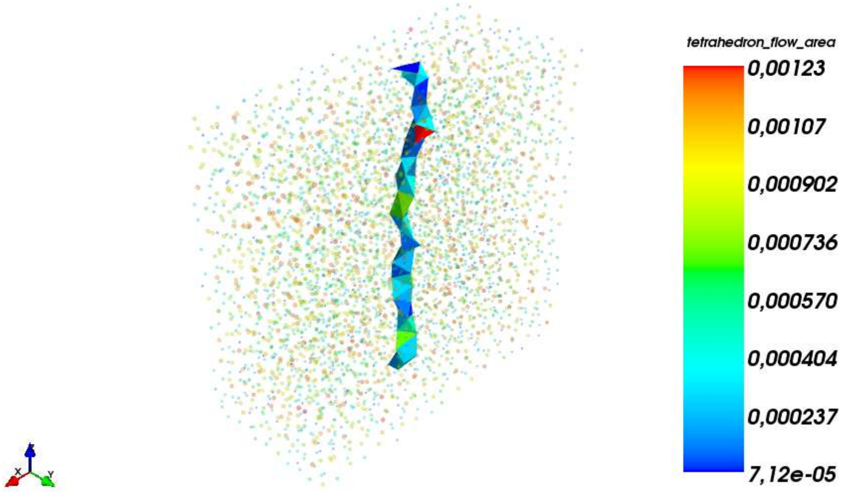
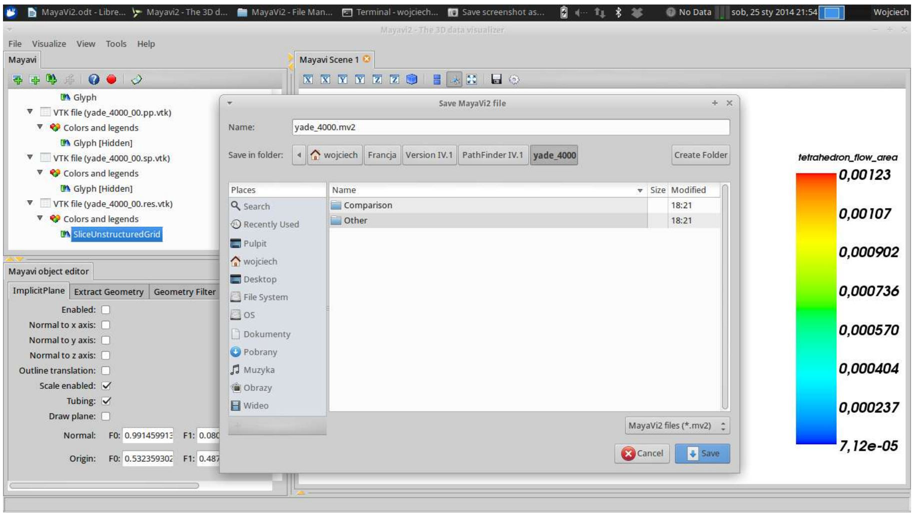


Fig. 6.31. Result of saving the current scene as a graphical file



Name: yade_4000.mv2

Save in folder: wojciech | Francja | Version IV.1 | PathFinder IV.1 | yade_4000

Enabled: □

Normal to x axis: □

Normal to y axis: □

Normal to z axis: □

Outline translation: □

Scale enabled: ☑

Tubing: ☑

Draw plane: □

Normal: F0: 0.99145991 F1: 0.080

Origin: F0: 0.532359302 F1: 0.487

Fig. 6.32. Saving the scene in a *.mv2 file

7 Additional software

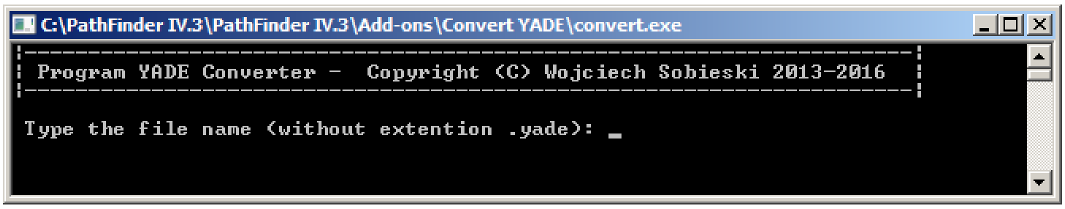
7.1 File converter from YADE to PathFinder

To export results from the YADE[9] software, perform following steps:

- Type in the YADE console: from yade import export
- Type in the YADE console: export.text("TheName.yade")

Next steps are as follows:

- Save or copy TheName.yade file from YADE to the Convert YADE directory in the main PathFinder folder.
- In Windows OS run the convert.exe file (it should be in the Convert YADE directory – if not, compile the source).
- In Unix/Linux OS, open console in the Convert YADE directory and type: ./convert.out (it should be in the Convert YADE directory, if not compile the source).
- Get TheName of the file exported in YADE (without the extension .yade) (Fig. 7.1).
- After the conversion, the input file TheName.in for PathFinder should be created.
- After the conversion, additional file TheName.in.vtk for bed visualisation should be available.


C:\PathFinder IV.3\PathFinder IV.3\Add-ons\Convert YADE\convert.exe

```
|----------------------------------------------------------------------|
| Program YADE Converter -  Copyright (C) Wojciech Sobieski 2013-2016  |
|----------------------------------------------------------------------|

 Type the file name (without extention .yade): _
```

Fig. 7.1. The YADE file format converter

The other possibility to use the YADE converter is runnig it with a call parameter as follows:

```
   convert.exe TheName
```

in Windows OS or

```
   ./convert.out TheName
```

in the Unix/Linux OS.

This possibility allows to run this file converter directly from the Python script. It may be done by performing following instructions

```
import subprocess
subprocess.call('./convert.out '+file, shell=True)
```

Additionally, the file converter creates a VTK file destined for visualisation of the bed.

The YADE file format converter is created in Fortran 90/95 language.

[9] The YADE GUI my be called from the console by using instruction: yade.qt.Controller()

In the Unix/Linux OS the output file must have rights to execution. For this purpose type in the shell console:

```
chmod +x convert.out
```

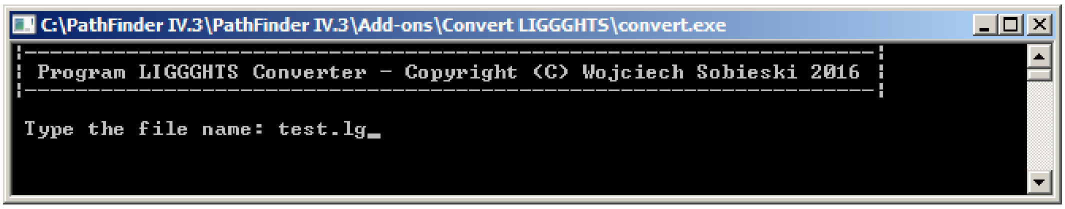
and next (to execute the file):

```
./convert.out
```

7.2 File converter from LIGGGHTS to PathFinder

In the LIGGGHTS code, the data export may have different forms. In the PathFinder package, an example of such results file is included. The User may follow, in which way the data may be converted. To run this exampe such steps must be doing:

- In Windows OS run the convert.exe file (it should be in the Convert LIGGGHTS directory – if not, compile the source).
- In Unix/Linux OS, open console in the Convert LIGGGHTS directory and type: ./convert.out (it should be in the Convert LIGGGHTS directory, if not compile the source).
- Get TheName of the file exported in LIGGGHTS (with the extension) (Fig. 7.2).
- After the conversion, the input file TheName.in for PathFinder should be created.
- After the conversion, additional file TheName.in.vtk for bed visualisation should be available.


C:\PathFinder IV.3\PathFinder IV.3\Add-ons\Convert LIGGGHTS\convert.exe

```
|-------------------------------------------------------------------|
| Program LIGGGHTS Converter - Copyright (C) Wojciech Sobieski 2016 |
|-------------------------------------------------------------------|

 Type the file name: test.lg_
```

Fig. 7.2. The LIGGGHTS file format converter

The other possibility to use the LIGGGHTS converter is runnig it with a call parameter as follows:

```
convert.exe TheName
```

in Windows OS or

```
./convert.out TheName
```

in the Unix/Linux OS.

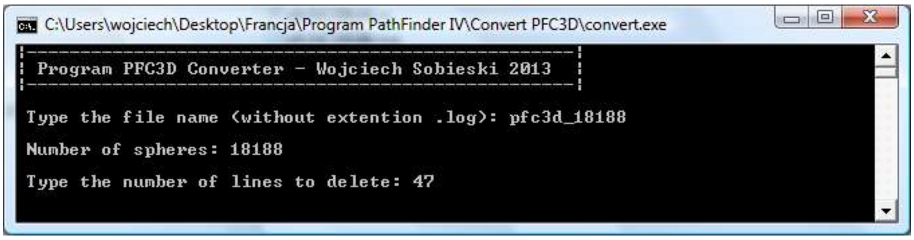
Important thing is that in the LIGGGHTS package, usually many files with data for one calculation are created. For this reason we recommend to use the default converter (named LPP) as the LIGGGHTS code; it is available on the same webpage. The Fortran code contained in the Add-Ons subfolder may then serve only as an additional possibility.

The LIGGGHTS file format converter is created in Fortran 90/95 language.

7.3 File converter from PFC^{3D} to PathFinder

In the PFC^{3D} code, the data export may have different forms. In the PathFinder package, an example of such results file is included. The PFC^{3D} converter shows the way to erase unnecessary lines in the file, to delete unnecessary signs and how to save the input file for PathFinder. Some intrinsic text functions of the Fortran programming language were used. The User probably will have to modify the code. In the example, the conversion process needs three variables: the name of the project, the number of particles in the bed and the number of unnecessary lines on the beginning the PFC^{3D} data file.

The PFC^{3D} file format converter is created in Fortran 90/95 language.


C:\Users\wojciech\Desktop\Francja\Program PathFinder IV\Convert PFC3D\convert.exe

```
|------------------------------------------------|
| Program PFC3D Converter - Wojciech Sobieski 2013 |
|------------------------------------------------|

Type the file name (without extention .log): pfc3d_18188

Number of spheres: 18188

Type the number of lines to delete: 47
```

Fig. 7.3. The PFC^{3D} file format converter

7.4 Characteristic dimension calculator

Characteristic dimension calculator is a small piece of software used for calculation of the characteristic dimension needed for the so-called Growing Particle Method available in the YADE DEM code. To use this software, a file with particle size distribution curve should be prepared (Fig. 7.4).

```
5.8598400e+00   0.0000000e+00
5.8761600e+00   8.4945735e-05
5.8924800e+00   3.5804112e-04
5.9088000e+00   1.1512195e-03
5.9251200e+00   3.2324246e-03
5.9414400e+00   8.1658647e-03
5.9577600e+00   1.8731095e-02
5.9740800e+00   3.9172232e-02
5.9904000e+00   7.4902028e-02
6.0067200e+00   1.3132521e-01
6.0230400e+00   2.1182373e-01
6.0393600e+00   3.1558203e-01
6.0556800e+00   4.3640887e-01
6.0720000e+00   5.6352779e-01
6.0883200e+00   6.8435463e-01
6.1046400e+00   7.8811293e-01
6.1209600e+00   8.6861145e-01
6.1372800e+00   9.2503463e-01
6.1536000e+00   9.6076443e-01
6.1699200e+00   9.8120556e-01
6.1862400e+00   9.9177079e-01
6.2025600e+00   9.9670423e-01
6.2188800e+00   9.9878544e-01
6.2352000e+00   9.9957862e-01
6.2515200e+00   9.9985171e-01
6.2678400e+00   9.9993666e-01
```

Fig. 7.4. An examplary content of a file with data with the particle distribution curve

The characteristic dimension is calculated as follows:

$$l_c = \sqrt[3]{\frac{V_s + V_p}{2}} \quad , \tag{7.1}$$

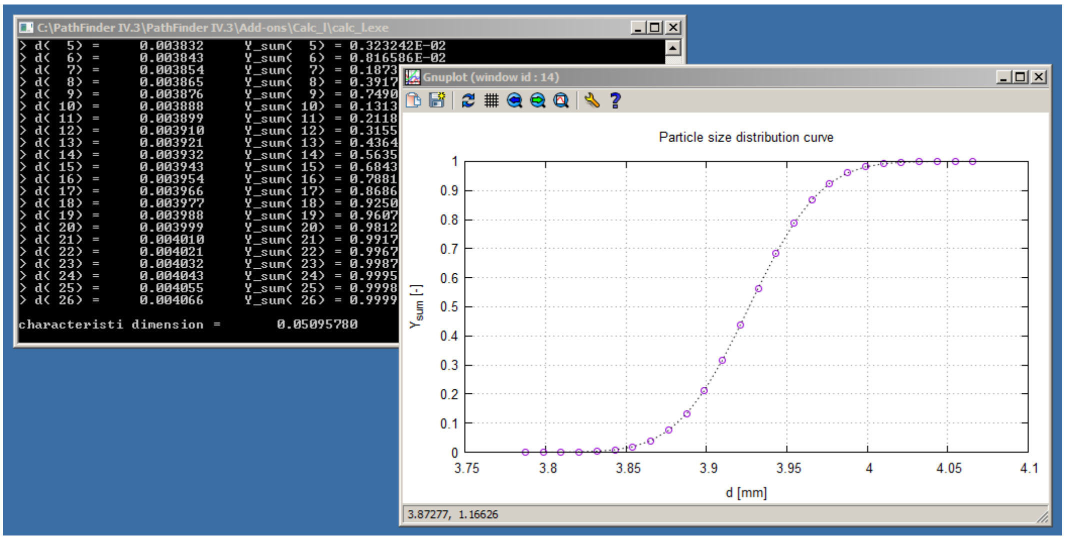
where

$$V_s = \sum_{i=1}^{n_{frac}} V_i^{\text{sum}} = \sum_{i=1}^{n_{frac}} n_i \cdot V_i = \sum_{i=1}^{n_{frac}} Y_i \cdot n_s \cdot V_i = \sum_{i=1}^{n_{frac}} Y_i \cdot n_s \cdot \frac{4}{3} \cdot \pi \cdot \left(\frac{d_i}{2}\right)^3 \tag{7.2}$$

and

$$V_p = \frac{e_{target}}{1 - e_{target}} \cdot V_s \quad . \tag{7.3}$$

Meaning of the symbols is as follows: n_{frac} - number of fractions in the virtual bed [-], V_i^{sum} - total volume of the particles of the i-th fraction [m^3], n_i - number of particles of the i-th fraction [-], V_i - volume of a single particle of the i-th fraction [m^3], Y_i - share of particles of the i-th fraction [-] (values from 0 to 1), n_s - total number of particles in the bed [-], d_i - particle diameter of the i-th fraction [m], e_{target} - target porosity of the bed [m^3/ m^3].



```
> d(  5) =      0.003832     Y_sum(  5) = 0.323242E-02
> d(  6) =      0.003843     Y_sum(  6) = 0.816586E-02
> d(  7) =      0.003854     Y_sum(  7) = 0.1873
> d(  8) =      0.003865     Y_sum(  8) = 0.3917
> d(  9) =      0.003876     Y_sum(  9) = 0.7490
> d( 10) =      0.003888     Y_sum( 10) = 0.1313
> d( 11) =      0.003899     Y_sum( 11) = 0.2118
> d( 12) =      0.003910     Y_sum( 12) = 0.3155
> d( 13) =      0.003921     Y_sum( 13) = 0.4364
> d( 14) =      0.003932     Y_sum( 14) = 0.5635
> d( 15) =      0.003943     Y_sum( 15) = 0.6843
> d( 16) =      0.003954     Y_sum( 16) = 0.7881
> d( 17) =      0.003966     Y_sum( 17) = 0.8686
> d( 18) =      0.003977     Y_sum( 18) = 0.9250
> d( 19) =      0.003988     Y_sum( 19) = 0.9607
> d( 20) =      0.003999     Y_sum( 20) = 0.9812
> d( 21) =      0.004010     Y_sum( 21) = 0.9917
> d( 22) =      0.004021     Y_sum( 22) = 0.9967
> d( 23) =      0.004032     Y_sum( 23) = 0.9987
> d( 24) =      0.004043     Y_sum( 24) = 0.9995
> d( 25) =      0.004055     Y_sum( 25) = 0.9998
> d( 26) =      0.004066     Y_sum( 26) = 0.9999

characteristi dimension =       0.05095780
```

Fig. 7.5. The l_c calculator

To use the l_c calculator, three call parameters must be defined:

```
calc_l.exe TheName ns e_target
```

in Windows OS or

```
./calc_l.out TheName ns e_target
```

in the Unix/Linux OS.

Important thing, is that the l_c calculator my be used directly from the Python script. It may be done through performing following instructions:

```
import subprocess
subprocess.call('./calc_l.out '+file+' '+str(n_s)+' '+str(e), shell=True)
```

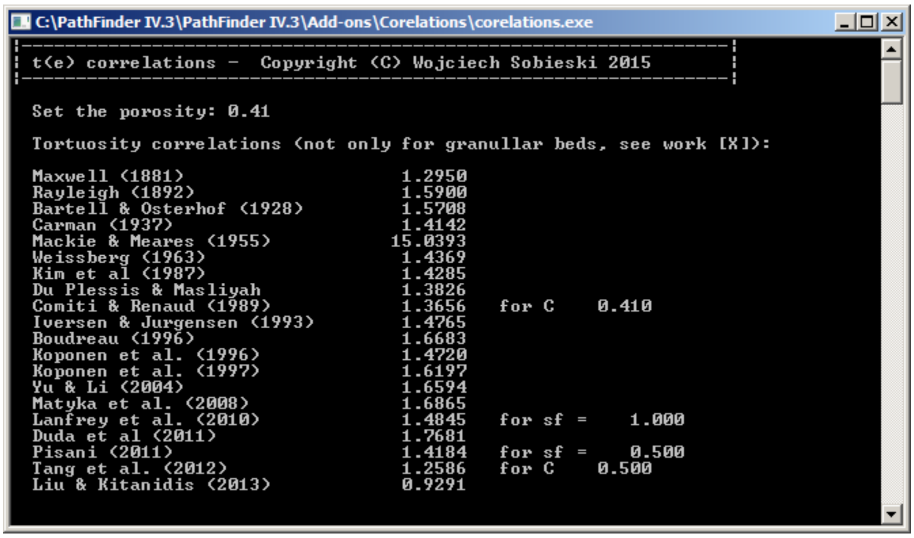
It results in a file with the value of the characteristic dimension, which is created in the same subfolder. This value may be read in the Python script with the use of the following instructions:

```
input = open(file+'.l_c','r')
try:
  for line in input:
    l = float(line[0:])
finally:
  input.close()
```

The l_c calculator was created in Fortran 90/95 language.

7.5 Porosity-tortuosity calculator

This tool allows to calculate different tortuosity values, but only when the porosity value is defined. Currently about 20 correlation formulas are collected. In some of them, a model constant must be defined and for this reason the User will need to modify and compile the source code. The gfortran compiler may be applied to achieve this aim.


C:\PathFinder IV.3\PathFinder IV.3\Add-ons\Corelations\corelations.exe

```
|--------------------------------------------------------------|
| t(e) correlations -  Copyright (C) Wojciech Sobieski 2015   |
|--------------------------------------------------------------|

 Set the porosity: 0.41

 Tortuosity correlations (not only for granullar beds, see work [X]):

 Maxwell (1881)                  1.2950
 Rayleigh (1892)                 1.5900
 Bartell & Osterhof (1928)       1.5708
 Carman (1937)                   1.4142
 Mackie & Meares (1955)         15.0393
 Weissberg (1963)                1.4369
 Kim et al (1987)                1.4285
 Du Plessis & Masliyah           1.3826
 Comiti & Renaud (1989)          1.3656   for C    0.410
 Iversen & Jurgensen (1993)      1.4765
 Boudreau (1996)                 1.6683
 Koponen et al. (1996)           1.4720
 Koponen et al. (1997)           1.6197
 Yu & Li (2004)                  1.6594
 Matyka et al. (2008)            1.6865
 Lanfrey et al. (2010)           1.4845   for sf =    1.000
 Duda et al (2011)               1.7681
 Pisani (2011)                   1.4184   for sf =    0.500
 Tang et al. (2012)              1.2586   for C    0.500
 Liu & Kitanidis (2013)          0.9291
```

Fig. 7.6. The porosity-tortuosity calculator

To use the porosity-tortuosity calculator, a call parameter must be defined as follows:

```
correlations.exe e
```

in Windows OS or

```
./correlations .out e
```

in the Unix/Linux OS.

If the call parameter is not given, the question about the porosity will be shown after running the program.

The results are visible in the console, as well as saved in a text file.

The porosity-tortuosity calculator was created in Fortran 90/95 language.

7.6 Ergun Air Calculator

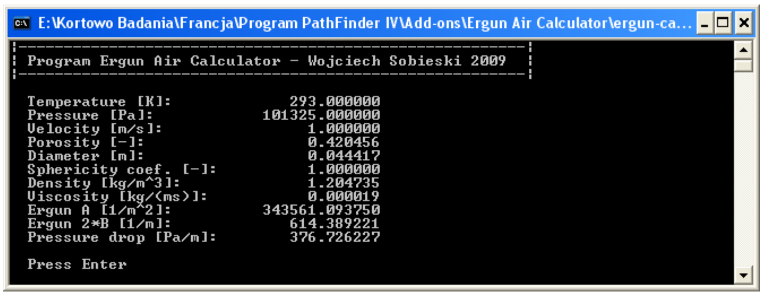
This small tool allows calculating values of the pressure drop in air flow through a porous bed using the Ergun equation

$$-\frac{dp}{dx}=\left[150\cdot\frac{(1-e)^2}{e^3\cdot(\varphi\cdot d)^2}\right]\cdot\left(\mu\cdot v_f\right)+\left[1.75\cdot\frac{(1-e)}{e^3\cdot(\varphi\cdot d)}\right]\cdot\left(\rho\cdot v_f^2\right)\quad , \tag{7.4}$$

where:

p - pressure [Pa],
x - a coordinate along which the pressure drop occurs [m],
e - porosity [-],
d -particle diameter [m],
φ - sphericity coefficient [-],
μ - dynamic viscosity coefficient of the fluid [kg/(m·s)],
ρ - density [kg/m^3],
v_f - filtration velocity [m/s].

All data is collected in the input file. The dynamic viscosity coefficient is calculated from the Sutherland's formula, the density in turn from the equation of state. At the end, a final report is shown (7.7) and saved to a file.


E:\Kortowo Badania\Francja\Program PathFinder IV\Add-ons\Ergun Air Calculator\ergun-ca...

```
¦----------------------------------------------------------¦
¦ Program Ergun Air Calculator - Wojciech Sobieski 2009    ¦
¦----------------------------------------------------------¦

 Temperature [K]:             293.000000
 Pressure [Pa]:            101325.000000
 Velocity [m/s]:                1.000000
 Porosity [-]:                  0.420456
 Diameter [m]:                  0.044417
 Sphericity coef. [-]:          1.000000
 Density [kg/m^3]:              1.204735
 Viscosity [kg/(ms)]:           0.000019
 Ergun A [1/m^2]:          343561.093750
 Ergun 2*B [1/m]:             614.389221
 Pressure drop [Pa/m]:        376.726227

 Press Enter
```

Fig. 7.7. The final report of the Ergun air calculator

In the Unix/Linux OS, the output file must have rights to execution. For this purpose type in the shell console:

```
chmod +x convert.out
```

and next (to execute the file):

```
./convert.out
```

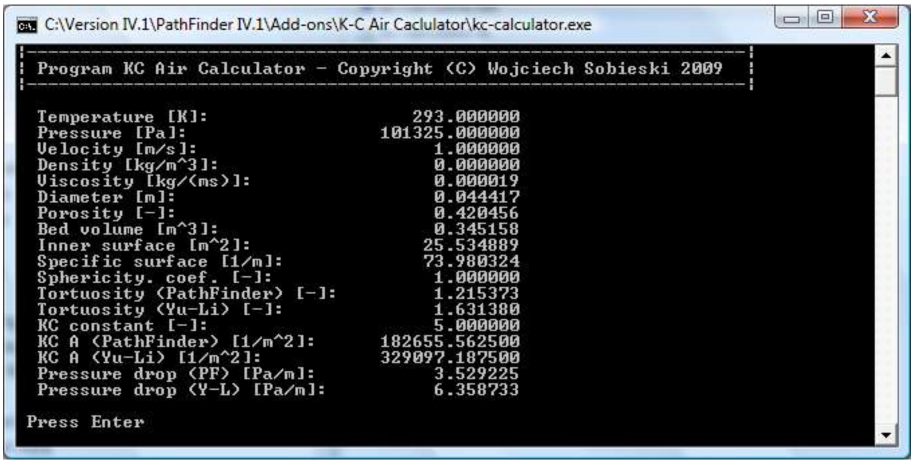
7.7 Kozeny-Carman Air Calculator

This small tool allows calculating values of the pressure drop in air flow through a porous bed using the Kozeny-Carman equation

$$-\frac{dp}{dx}=C_{KC}\cdot\tau_f\cdot S_0^2\cdot\frac{(1-e)^2}{e^3}\cdot(\mu\cdot v_f) \quad , \tag{7.5}$$

where:

p - pressure [Pa],
x - a coordinate along which the pressure drop occurs [m],
C_{KC} - a model constant (so called Kozeny-Carman pore shape factor) [-],
τ_f - the tortuosity factor [m^2/m^2] defined as the square of the tortuosity τ [m/m],
S_0 - the specific surface of the porous body [1/m],
e - the porosity [-],
μ - dynamic viscosity coefficient of the fluid [kg/(m·s)],
v_f - filtration velocity [m/s].


C:\Version IV.1\PathFinder IV.1\Add-ons\K-C Air Caclulator\kc-calculator.exe

```
|-----------------------------------------------------------------------|
| Program KC Air Calculator - Copyright (C) Wojciech Sobieski 2009       |
|-----------------------------------------------------------------------|

 Temperature [K]:                          293.000000
 Pressure [Pa]:                         101325.000000
 Velocity [m/s]:                             1.000000
 Density [kg/m^3]:                           0.000000
 Viscosity [kg/(ms)]:                        0.000019
 Diameter [m]:                               0.044417
 Porosity [-]:                               0.420456
 Bed volume [m^3]:                           0.345158
 Inner surface [m^2]:                       25.534889
 Specific surface [1/m]:                    73.980324
 Sphericity. coef. [-]:                      1.000000
 Tortuosity (PathFinder) [-]:                1.215373
 Tortuosity (Yu-Li) [-]:                     1.631380
 KC constant [-]:                            5.000000
 KC A (PathFinder) [1/m^2]:             182655.562500
 KC A (Yu-Li) [1/m^2]:                  329097.187500
 Pressure drop (PF) [Pa/m]:                  3.529225
 Pressure drop (Y-L) [Pa/m]:                 6.358733

Press Enter
```

Fig. 7.8. The final report of the Kozeny-Carman air calculator

In the program, the tortuosity according to the Yu and Li theory is additionally calculated. For this reason, all next equations are calculated two times: once for the assumed tortuosity (e.g. obtained with the PathFinder software) and once for the Yu and Li tortuosity.

All data is collected in the input file. The dynamic viscosity coefficient is calculated from the Sutherland's formula. At the end, a final report is shown (7.7) and saved to a file.

In the Unix/Linux OS, the output file must have rights to execution. For this purpose type in the shell console:

```
chmod +x convert.out
```

and next (to execute the file):

```
./convert.out
```

7.8 The Sensitivity Calculator

This tool shows dependencies between the result and changes in input parameters for Ergun and

Kozeny-Carman laws. The input data files may be created manually or copied from the Comparison directory in the PathFinder project folder. Such files are generated automatically by the PathFinder code in every calculation.

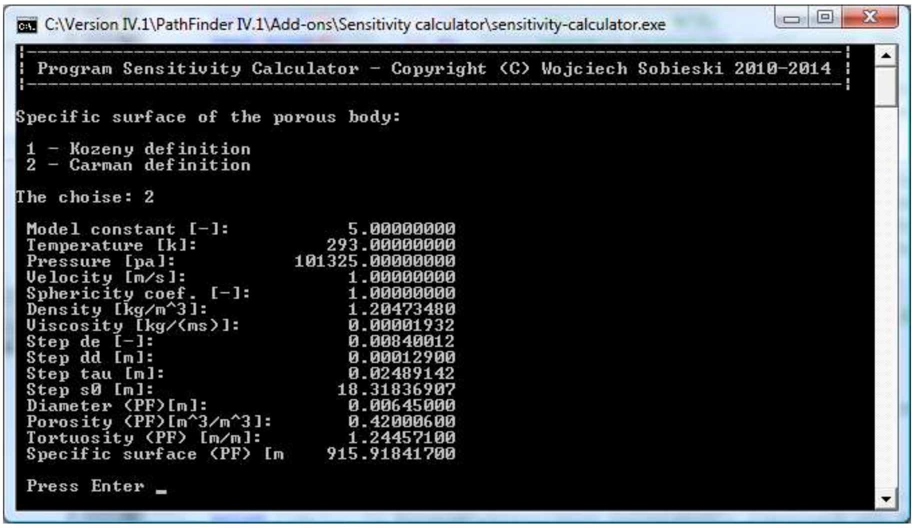
After running the program, a question concerning the definition of the specific surface of the porous body is shown (Fig. 7.9). Calculations are available after choosing the Kozeny or Carman approach. Results of calculations are shown (and saved) in graphical form. Appropriate Gnuplot scripts are included in the Sensitivity Calculator folder. An example of such visualization is shown in Fig. 7.10.

In the Unix/Linux OS, the output file must have rights to execution. For this purpose type in the shell console:

```
chmod +x convert.out
```

and next (tu execute the file):

```
./convert.out
```



C:\Version IV.1\PathFinder IV.1\Add-ons\Sensitivity calculator\sensitivity-calculator.exe

```
|---------------------------------------------------------------------------|
| Program Sensitivity Calculator - Copyright (C) Wojciech Sobieski 2010-2014 |
|---------------------------------------------------------------------------|

Specific surface of the porous body:

 1 - Kozeny definition
 2 - Carman definition

The choise: 2

 Model constant [-]:              5.00000000
 Temperature [k]:               293.00000000
 Pressure [pa]:              101325.00000000
 Velocity [m/s]:                  1.00000000
 Sphericity coef. [-]:            1.00000000
 Density [kg/m^3]:                1.20473480
 Viscosity [kg/(ms)]:             0.00001932
 Step de [-]:                     0.00840012
 Step dd [m]:                     0.00012900
 Step tau [m]:                    0.02489142
 Step s0 [m]:                    18.31836907
 Diameter (PF)[m]:                0.00645000
 Porosity (PF)[m^3/m^3]:          0.42000600
 Tortuosity (PF) [m/m]:           1.24457100
 Specific surface (PF) [m       915.91841700

 Press Enter _
```

Fig. 7.9. Choosing the definition of the specific surface of the porous body

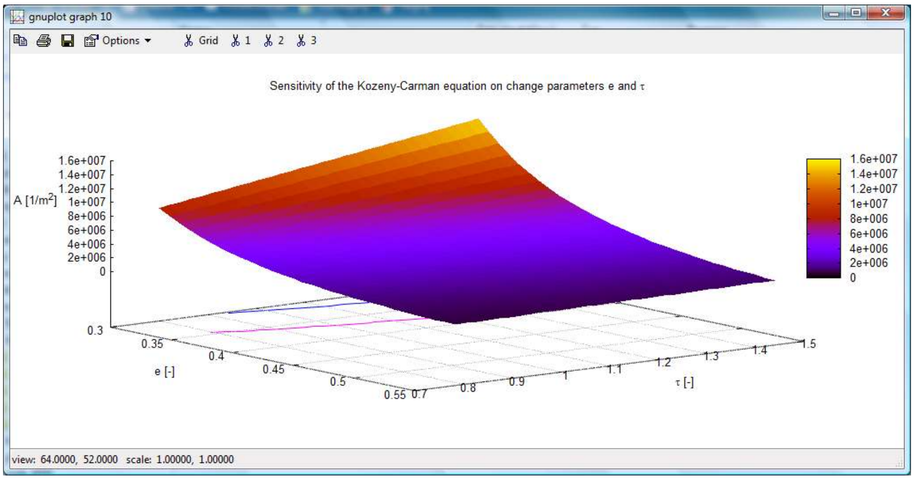


Fig. 7.10. An example of visualization a model sensitivity

8 Publications

1. Sobieski W., Trykozko A.: Darcy's and Forchheimer's laws in practice: Part 2. The numerical model. TECHNICAL SCIENCES, Vol. 4 (2014), pp. 337-350. ISSN 1505-4675.
2. Sobieski W., Trykozko A.: Darcy's and Forchheimer's laws in practice: Part 1. The experiment. TECHNICAL SCIENCES, Vol. 4 (2014), pp. 221-335. ISSN 1505-4675.
3. Sobieski W., Zhang Q.: Sensitivity analysis of Kozeny-Karman and Ergun equations. TECHNICAL SCIENCES, Vol. 3 (2014), pp. 235-248.
4. Dudda W., Sobieski W.: Modification of the PathFinder algorithm for calculating granular beds with various particle size distributions. TECHNICAL SCIENCES, Vol. 17, No 2 (2014), pp. 135-148.
5. Sobieski W.: The quality of the base knowledge in scientific investigations. Chapter in the monograph "Scientific Researches in the Department of Mechanics and Machine Design". Olsztyn 2014 (in Polish).
6. Sobieski W., Dudda W.: Sensitivity Analysis as a Tool for Estimating Numerical Modeling Results. DRYING TECHNOLOGY, Vol. 32, No. 2 (2014), pp. 145-155.
7. Sobieski W., Lipiński S.: PathFinder User's Guide. University of Warmia and Mazury in Olsztyn (Poland), 2013.
8. Sobieski W.: The basic closures of fluid mechanics in form characteristic for the finite volume method. TECHNICAL SCIENCES, Vol. 16, No. 2 (2013), pp. 93-107.
9. Sobieski W., Zhang Q., Liu, C.: Predicting tortuosity for airflow through porous beds consisting of randomly packed spherical particles. TRANSPORT IN POROUS MEDIA, Vol. 93, No. 3 (2012), pp. 431-451.
10. Sobieski W., Trykozko A.: Sensitivity aspects of Forchheimer's approximation. TRANSPORT IN POROUS MEDIA, Vol. 89, No. 2 (2011), pp 155-164.
11. Sobieski W.: Possibility to use the diffusion equation to heat flow modeling in a composting process. JOURNAL OF APPLIED COMPUTER SCIENCE, Vol. 19, No. 2 (2011), pp. 111-123.
12. Sobieski W.: The basic equations of fluid mechanics in form characteristic of the finite volume method. TECHNICAL SCIENCES, Vol. 14, No. 2 (2011), pp. 299-313.
13. Peszyńska M., Trykozko A., Sobieski W.: Forchheimer law in computational and experimental studies of flow through porous media at porescale and mesoscale. GAKUTO International Series, MATHEMATICAL SCIENCES AND APPLICATIONS, Vol. 32, pp. 463-482, Tokyo, Japan 2010.
14. Sobieski W.: Examples of using the Finite Volume Method for modeling fluid-solid systems. TECHNICAL SCIENCES, Vol. 13 (2010), pp. 256-265.
15. Sobieski W.: Selected aspects of developing a simulation model of a spouted bed grain dryer based on the Eulerian Multiphase Model. DRYING TECHNOLOGY, Vol. 28, No. 12 (2010), pp. 1331-1343.
16. Liu, C. Zhang Q., Chen Y.: Discrete element simulations of vibration characteristics of bulk grain in storage bins. Transactions of the ASABE, Vol. 53, No. 5 (2010), pp. 1653-1659.
17. Sobieski W.: Use of numerical models in validating experimental results. JOURNAL OF APPLIED COMPUTER SCIENCE, Vol. 18, No. 1 (2010), pp. 49-60.
18. Sobieski W.: Calculating tortuosity in a porous bed consisting of spherical particles with known sizes and distribution in space (in Polish). Research report 1/2009, Winnipeg (Canada), 2009.
19. Sobieski W.: GNU Fortran with data visualization elements (in Polish). University of Warmia and Mazury, Olsztyn 2008, 403 pages.
20. Liu, C., Zhang Q., Chen Y.: PFC3D simulations of lateral pressures in model bins. Paper No 083340 (2008). American Society of Agricultural and Biological Engineers, St. Joseph, MI. 13 pp.
21. Liu, C., Zhang Q., Chen Y.: PFC3D simulations of vibration characteristics of bulk solids in storage bins. Paper No 083339 (2008), American Society of Agricultural and Biological Engineers, St. Joseph, MI. 14 pp.
22. Sobieski W.: The quality of the base knowledge in a research process. In "Scientific researches in the department of mechanics and machine design" 2014, 2, 29-47. ISBN 9788392978114. University of Warmia and Mazury in Olsztyn, Poland (in Polish).
23. Sobieski W., Dudda W., Lipiński S.: A new approach for obtaining the geometric properties of a granular porous bed based on DEM simulations. TECHNICAL SCIENCES (in press).
24. Sobieski W., Lipiński S.: The porosity-tortuosity correlations for granular beds. TECHNICAL SCIENCES (in press).