

TOUGH2Path Manual

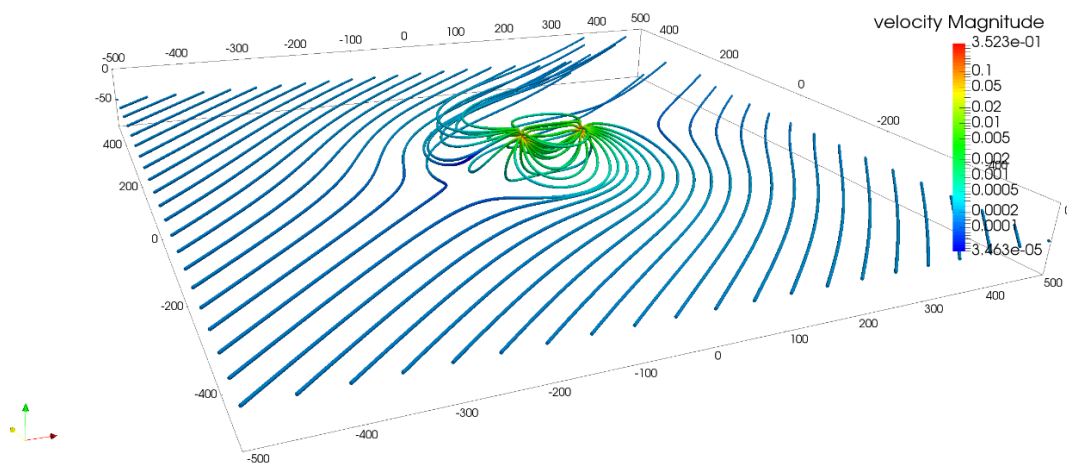
S. Bonduà, V. Bortolotti

DICAM

University of Bologna

Ver. 1.0

<http://software.dicam.unibo.it/tough2path>



Sommario

1	Introduction.....	3
1.1	TOUGH2Path Input files	3
1.2	TOUGH2Path output files	4
2	Note on TOUGH2Viewer.dat	5
3	Usage	5
4	Examples.....	6
4.1	2D_Regular	6
4.1.1	TOUGH2 Simulation result visualization.....	6
4.1.2	TOUGH2Path computation results	9
4.2	2D_45Regular	12
4.3	2D_SemiRegular	12
4.4	2D_Voronoi.....	12
4.5	2D_CVT	13
4.6	3D Models.....	13
5	References	17

1 Introduction

TOUGH2Path can compute the path-line of a particle thorough each block of a 3D Voronoi discretized domain using directly the output file of a TOUGH2 (Pruess et al., 1999) simulation and using the Klausen et al. (2012) method. TOUGH2Path is a command line software coded in C++ running under Windows. It uses a parallel programming approach using OpenMP (Open Multi-Processing)¹ for speed-up computation, and so permitting the contemporaneously computation of a high number of particle paths with a reasonable computation time cost.

1.1 TOUGH2Path Input files

TOUGH2Path needs in input the geometrical information about the grid blocks shape and flows data between blocks. In particular, as the structure of the TOUGH2 output file depends on the specific module of Equation of State used (see Pruess et al. (1999) for details), TOUGH2Path needs to be instructed about where are the positions of the variables to be read (see Figure 1 for details).

Therefore, the input data of TOUGH2Path is composed of several files:

- TOUGH2path.par: contains the parameters of the path-particle computation, as the step of integration, maximum number of iterations, directive for ending computation, domain limits, etc. see Appendix D for details.
- MESH: is the standard MESH file of TOUGH2 (see TOUGH2 manual, Pruess et al. (1999), Finsterle (2012)). Elements with volume bigger than $1.0 \cdot 10^{20}$ are considered “ending block”, so when the particle is entering in a ending block the computation will terminate;
- TOUGH2Viewer.dat: is the file as generated by VORO2MESH (see Bonduà et al. 2017, 2012) containing the geometrical information about each block (vertexes, edges, faces and faces normal);
- T2.out: the standard output file of TOUGH2. It must contain flows velocity between blocks (usually named VEL(LIQ) (m/s) or VEL(GAS) (m/s));
- header.dat: is a file containing at least 3 lines of characters, like they appear in the T2.out file. This is to allow TOUGH2Path to have an easy parsing of the TOUGH2 output files (that are different for each EOS module), the user have to specify the position of the ELEM1, ELEM2 and of the variables to be read (see Figure 1 for details);
- seed_points.dat: an ASCII file, space separated x, y, z coordinate of the initial position of the particles set on which calculate the path;
- ending_points.dat: an ASCII file, space separated x, y, z, r (radius) coordinate of the ending position of the particles set on which terminate the path computation. The number of ending points have to be the same of the seed points presents in the file seed_points.dat. The computation of the particle path will be terminated when the particle 3D distance between particle position and respective ending point is lower than r.
- GENER: is the standard GENER file of TOUGH2. If the specified flow rate of a GENER element is greater than zero (injection block), then a seed point for each face is added to the seed points set. If the flow rate is lower than zero (production block), then the block will be considered as an ending block.

ELEM1	ELEM2	INDEX	FHEAT	FLOH/FLOF	FLOF	FLO (NaCl)	FLO (CO2)	FLO (GAS)	FLO (LIQ.)	VEL (GAS)	VEL (LIQ.)
			(W)	(J/KG)	(KG/S)	(KG/S)	(KG/S)	(KG/S)	(KG/S)	(M/S)	(M/S)
11111	22222										+++++++

¹ OpenMP Architecture Review Board, The OpenMP Specification, <http://www.openmp.org>, 2017 (accessed: 29.09.2017).

A11	1	A21	1	1	-.281857E+05	0.103545E+06	-.272207E+00	-.103929E-07	0.000000E+00	0.000000E+00	-.272207E+00	0.000000E+00	-.72695E-06
-----	---	-----	---	---	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	-------------

Figure 1 - Example of the header.dat file. The 3-rd line (highlighted in yellow) is edited by the user. The “11111” and “22222” strings are used for column identification of the blocks name, while the “+++++” string is used for column identification of the flow variable to be read.

1.2 TOUGH2Path output files

The standard output of TOUGH2Path is composed of a set of ASCII files with information about the path of each input particle.

Specifically, the output files of TOUGH2Path are:

- For each tracked particle a path-file is created. The path file is an ASCII space separated file, where each line contains: the position of the particle (x, y, z), the 3 component of the velocity (vx, vy, vz), the ID of the block crossed, the travelled distance, the integration time. The naming convention of these file is: path[particle number].dat;
- T2Path.vtp file: a XML (polydata Paraview² file) file for the particles pathlines visualization in Paraview. The file contains all the information for the visualization of the particle path as a polyline:
 - the coordinate each point (x,y,z);
 - the estimated velocity (vx, vy, vz);
 - the ID of the block that the point belongs to;
 - the integration time;
 - the total travelled distance;
 - the reason for the path line computation termination (a code between 1 and 8, see Table 1).
- Node_vel files: are comma separated file (csv) of the velocity calculated for each node blocks for each time step. The naming convention of these file is: Node_vel_[time step].csv.
- TOUGH2Path.log: a file containing information about parameters used, time of computation warnings and, if any, errors during the computation.

Reason for termination	Code
Particle goes in an ending block	1
Estimated velocity < v_min	2
n_iter > max_iter	3
Total_time > time_max	4
The distance between the particle and an ending point is lower than r	5
Particle position is offside of the particle-path computation domain defined by x_min, x_max, etc. (can be smaller than the discretized domain of the TOUGH2 Simulation)	6
Reserved code	7
Particle leave the domain	8
Table 1 - Reasons for program stops and corresponding code used by TOUGH2Path.	

² www.paraview.org, Accessed [Accessed 2018-01-18]

2 Note on TOUGH2Viewer.dat

In order to run properly TOUGH2Path, the file TOUGH2viewer.dat is necessary. You can generate this file in two ways: (i) using voro2mesh (<http://software.dicam.unibo.it/voro2mesh>); (ii) using TOUGH2Viewer (<http://software.dicam.unibo.it/tough2viewer>).

If you created your grid by using VORO2MESH, by default the file tough2viewer.dat was generated and is present in your working folder.

If you created your MESH file with the built-in grid generator MESHMAKER of TOUGH2, you can load your model in TOUGH2Viewer as “Structured grid” and then use the new function “generate TOUGH2Viewer.dat file” using the command button of the input box dialog window).

For grid generated with other software, you need to re-generate your MESH file using VORO2MESH (or MESHMAKER in case of regular grid) and re-run your TOUGH2 simulation.

Important note: when using voro2mesh for generating the grid and for computing the particle path line using TOUGH2Path, use in the input file of voro2mesh (voro.par) the option:

write_tough2viewer_dat=2.

With this option, the block geometry of the blocks (node, vertex and normals) is written in high precision format (c++, “%.17f”), needed for accurate path line tracing by TOUGH2Path.

3 Usage

TOUGH2Path installation consist of coping the tough2path.exe file in folder. In this example we choose “d:\executables”. Add the “d:\executables” to your environment variable path Operative System settings.

TOUGH2Path was created with VS2015 and dynamically linked, so you may need to copy the “vcomp140.dll” and other dynamic link library files in to your “d:\executables” folder (a warning message will appear and abort execution of the program in case of missing dll).

Once the previous operation are completed, open a command dos shell, move to a example folder and simply type tough2path [enter].

4 Examples

Download and extract the tough2_example.zip file from the <http://software.dicam.unibo.it/tough2path> internet website.

Move to a folder, open a command dos window and type tough2path followed by [enter].

To visualize the TOUGH2Path results, open the T2Path.vtp (and if needed, the other created files) file with Paraview.

To visualize the numerical simulated model jointly with the computed pathlines you need of the out.vtu file, that can be created by using the TOUGH2Viewer software (see TOUGH2Viewer manual for details).

The example file folder contain two main sub folder, named 2D and 3D. The example are related to a simulation of a isothermal injection/production scenario, for a 2D case and for a 3D case. The 2D scenario is simulated using 5 grid type, while the 3D case by using 6 grid type.

The grids are obtained by generating the nodes of the blocks of the grid in a domain of $500 \times 500 \times 300 \text{ m}^3$ for the 2D case and $500 \times 500 \times 500 \text{ m}^3$ for the 3D case. The domain is discretized using VORO2MESH (Bonduà et al. 2017). VORO2MESH allow automatically generation of the MESH and the tough2viewer.dat file, needed the former for the numerical simulation with TOUGH2 and the latter for the visualization using TOUGH2Viewer and for the pathline computation using TOUGH2Path. A detailed step by step procedure is given for the computation and visualization of the 2D Regular grid. To compute and visualize the other case studies, simply repeat the procedure as explained for the 2D Regular Case.

4.1 2D_Regular

The grid of the 2D_Regular is shown in Figure 2.

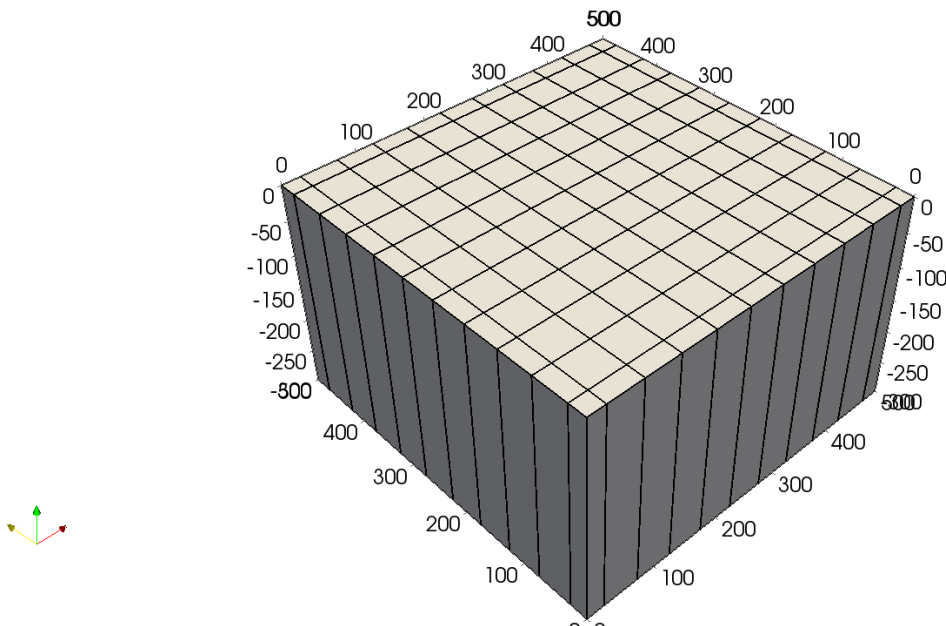


Figure 2 – The 2D_regular grid.

4.1.1 TOUGH2 Simulation result visualization

The visualization of the TOUGH2 simulation results can be performed using TOUGH2Viewer and Paraview. From the TOUGH2Viewer file menu, choose File->Open unstructured V++ grid ...

Select the tough2viewer.dat, the MESH and the t2.out file by clicking the respective buttons. Once all files are loaded, click the Export2VTU command button, as shown in Figure 3.

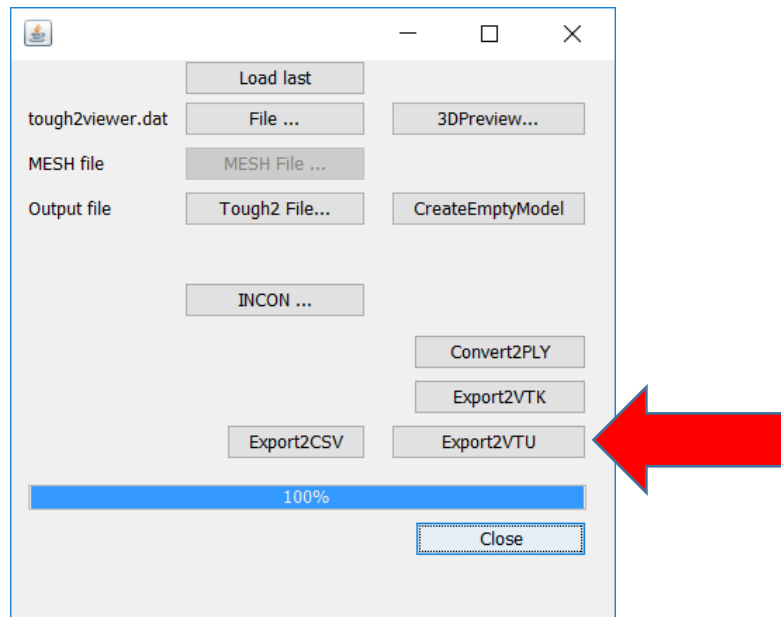




Figure 3 – Input unstructured V++ grid dialog window of TOUGH2Viewer

A set of out_[i].vtu file will be generated.

From the File menu of Paraview, choose File->Open ... and select out..vtu file. Note that Paraview allow the loading of timeseries data, by reading files progressively numerated. If you open the out.pvd (created by TOUGH2Viewer too), then also the information related to the time of the simulation print will available in Paraview. Depending on your default parameters setting of Paraview, you may modify several properties to achieve the visualization results of the Figure 5. Basically, you have to change several properties. In the pipeline Browser modify the properties as showed in Figure 4. Set the representation to “surface with edge”, line thickness=2, color of edge to “black”. Click the  and set the color scale by selecting the classical color combination “Blue to Rainbow”. The model will appear as in Figure 5



To move through time steps, click the  command button frame navigation bar. To automatically print all the time step choose from the File menu the File->Save animation and choose the jpg file format.

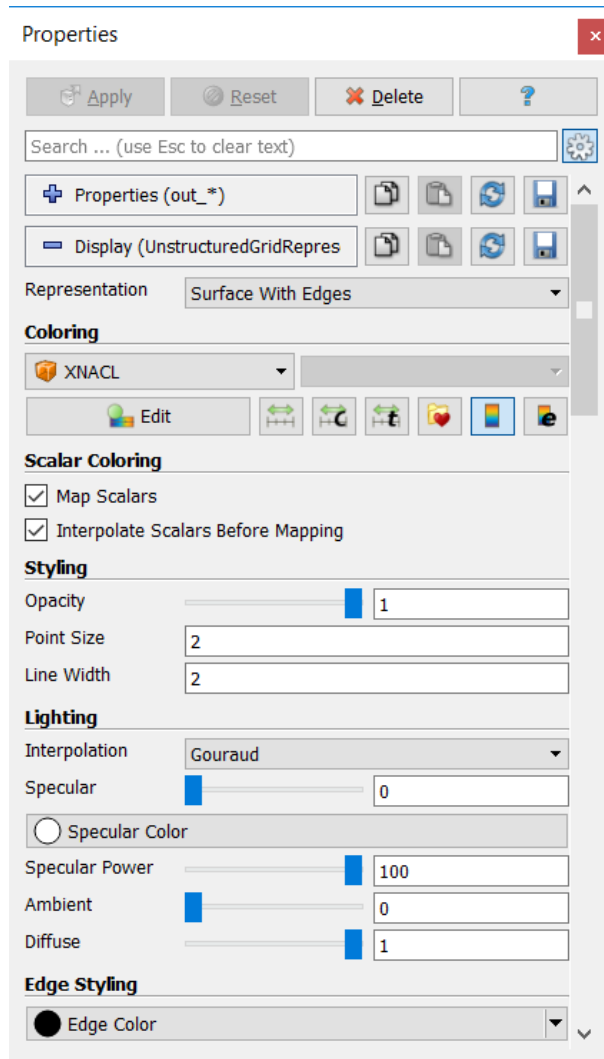
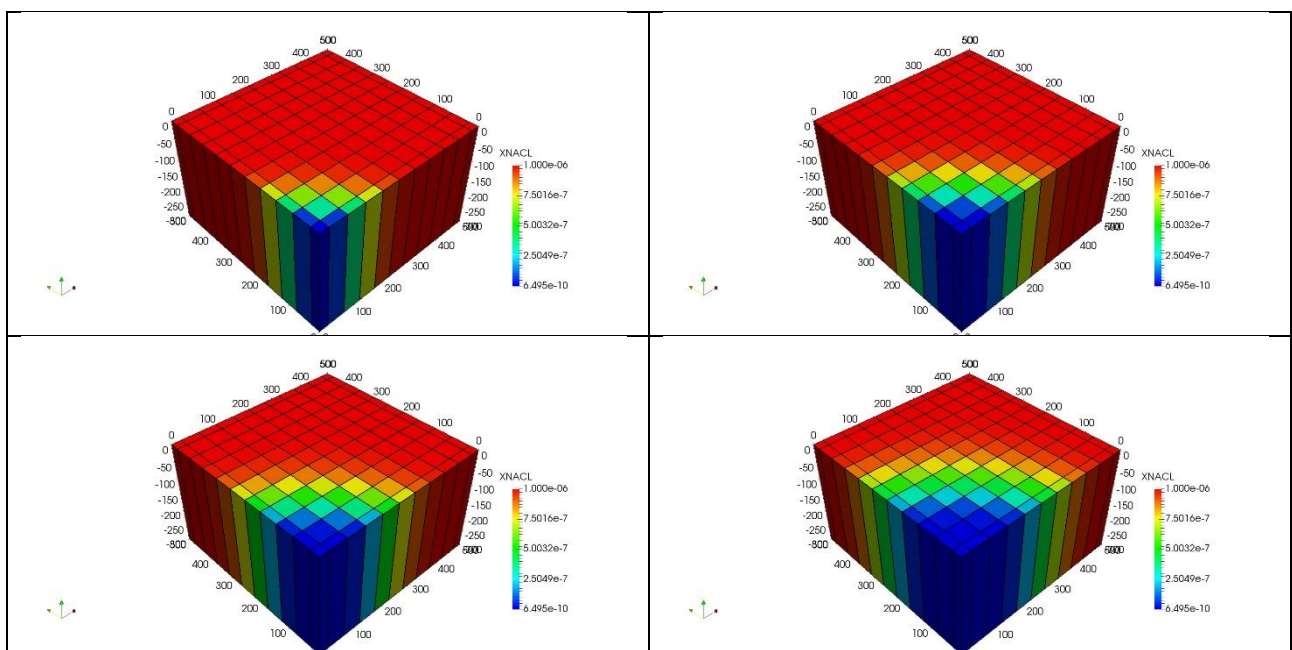


Figure 4 – Pipeline browser properties



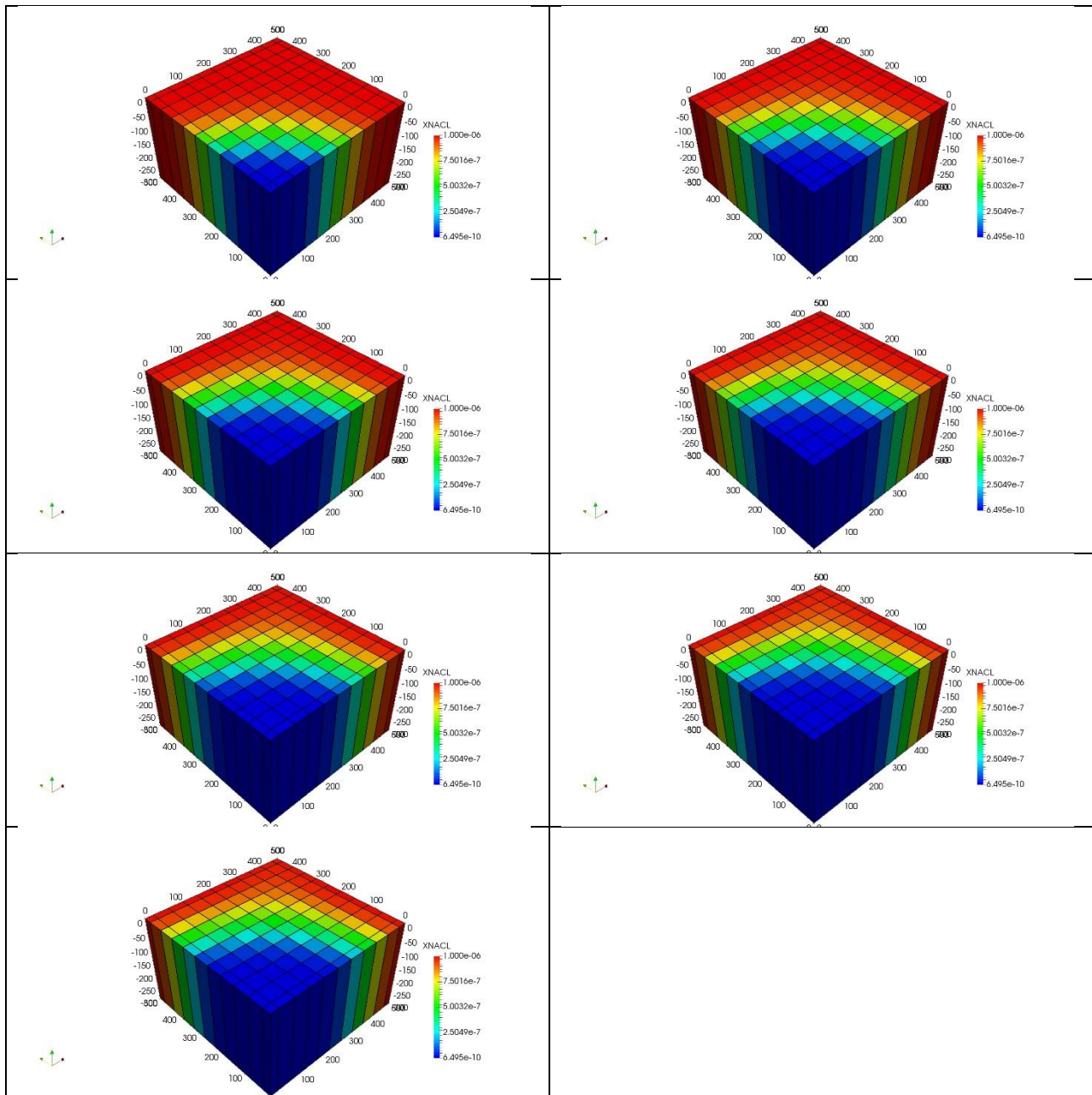


Figure 5 – Visualization of the 11 TOUGH2 simulated time step for the XNaCl saturation. The initial XNaCl saturation is 10^{-6} , while the NaCl content of the injected fluid is zero (pure water).

4.1.2 TOUGH2Path computation results

The file needed for the TOUGH2Path computation are already prepared in the 2D_Regular_4_particles folder.

Run tough2path in the 2D_Regular_4_particles. folder

After TOUGH2Path finished you will find all files needed for visualization with Paraview (or other software of your choice).

From the file menu of Paraview, select File->Open and choose T2Path.vtp file, the file containing TOUGH2Path computation results. Note that the pathlines are not visible as they fall completely inside the numerical model. From the Pipeline Browser of Paraview, select the Out_*. object and choose Representation as “wireframe”. Select now the T2Path object and choose as the coloring properties, the

Velocity (or other variables of your choice) and thickness of the line to 4. The model will now become as shown in Figure 6.

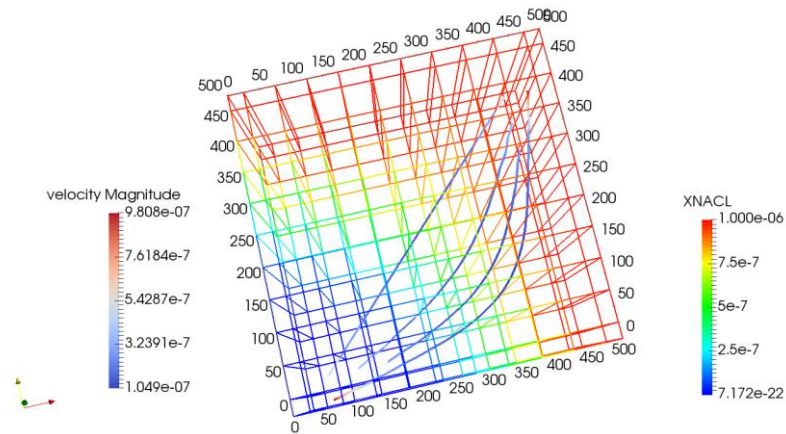


Figure 6 – Visualization of the model of the pathlines.

To draw vectors about the versus of the particle movement along the pathlines (keyword activating TOUGH2Path for the writing of this file: `print_path_line_vectors=1, n_vector_for_each_path_line=4` of the TOUGH2Path.par file), open the file Tough2_path_line_vectors.csv.vtk as VTK Legacy file and apply a Glyph filter (cone). Adjust dimensions (in the example of Figure 7, cone radius=0.1, cone height=0.2), color (set to GlyphVector) and Masking se to “all points” to obtain the representation of Figure 7.

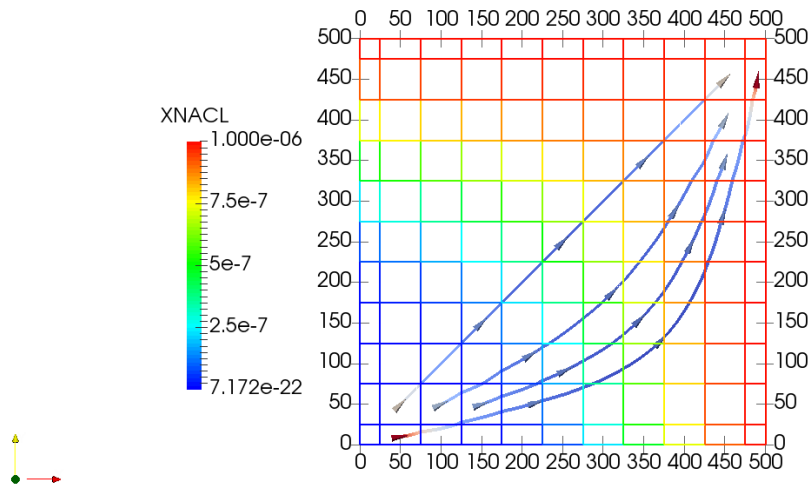


Figure 7 – Versus of the particle movement along their pathlines.

An enhancement of the visualization may be obtained by representing the pathline a “tubes”. To do that, select the T2Path object from the Pipeline Browser of Paraview and from the Filters->Alphabetical->Tube and apply changes.

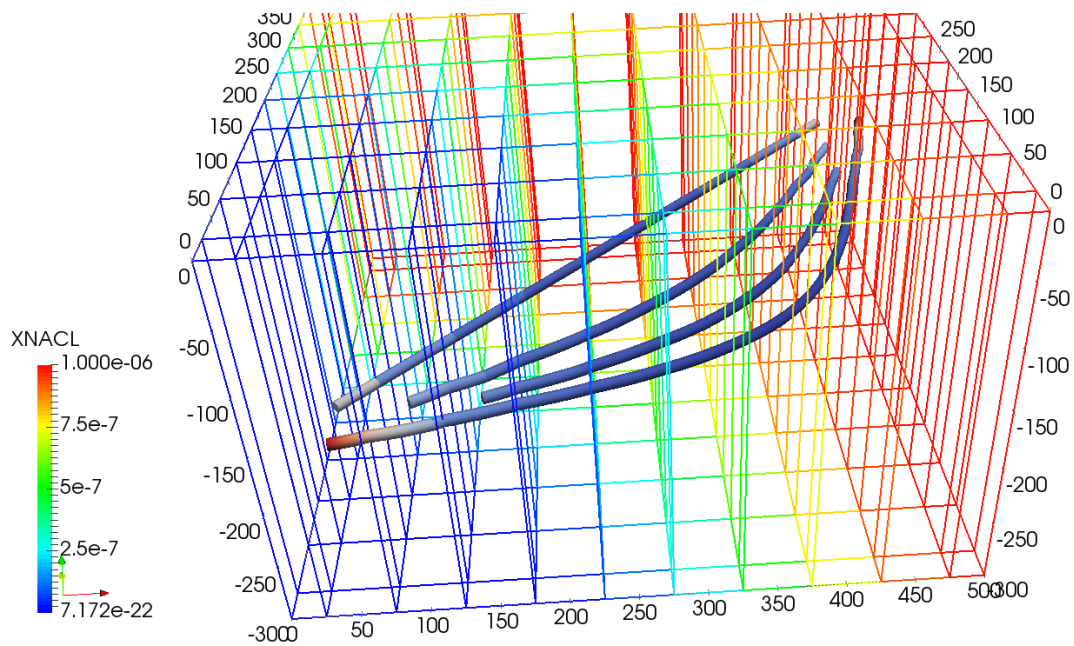


Figure 8 – Visualization of the pathline as tubes

TOUGH2Path compute optionally (`print_node_vel=1` of the `tough2path.par` file) the node velocity for all grid nodes and for all time steps of the TOUGH2 output file. To visualize the flow vector field, open the `node_vel.vtk` file from File->Open... as VTK legacy file.

After applying changes, select the `Node_vel*` object and apply the filter Filter->Alphabetical->Glyph.

Change the coloring option to “GlyphVector” and Masking to “all points” to obtain a representation similar to the Figure 9.

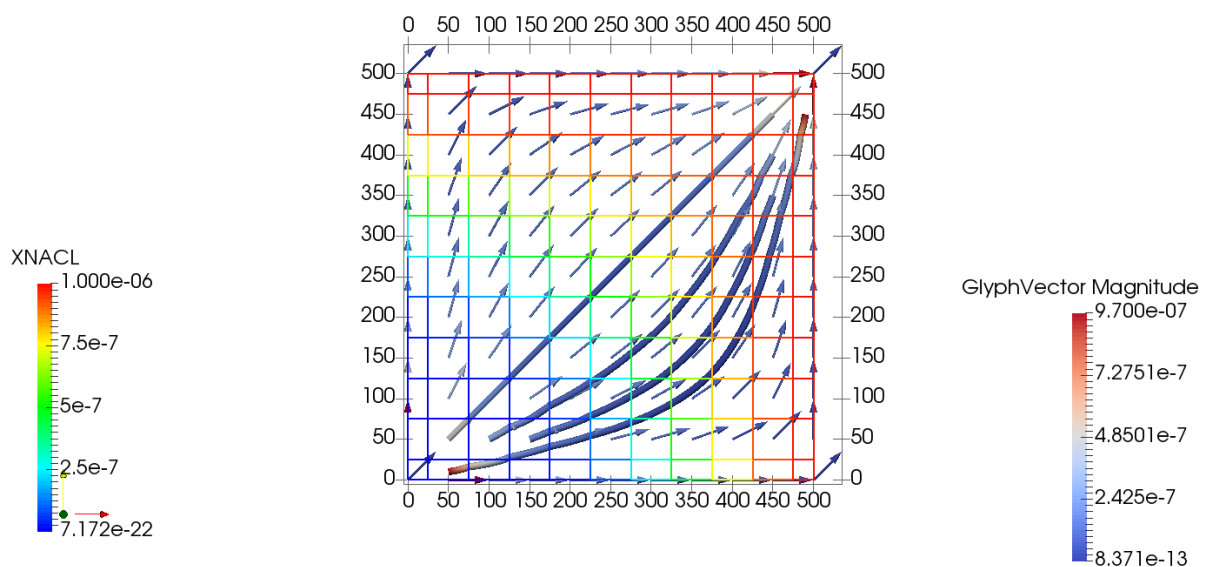


Figure 9 – Nodal flow Vector representation (2D TopView)

4.2 2D_45Regular

Results visualization are shown in Figure 10.

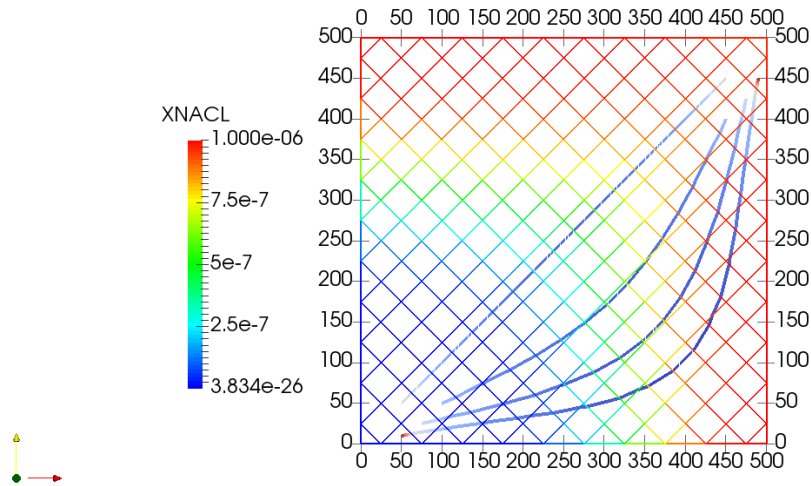


Figure 10 – Result visualization for the 2D_45Regular case.

4.3 2D_SemiRegular

Results visualization are shown in Figure 11.

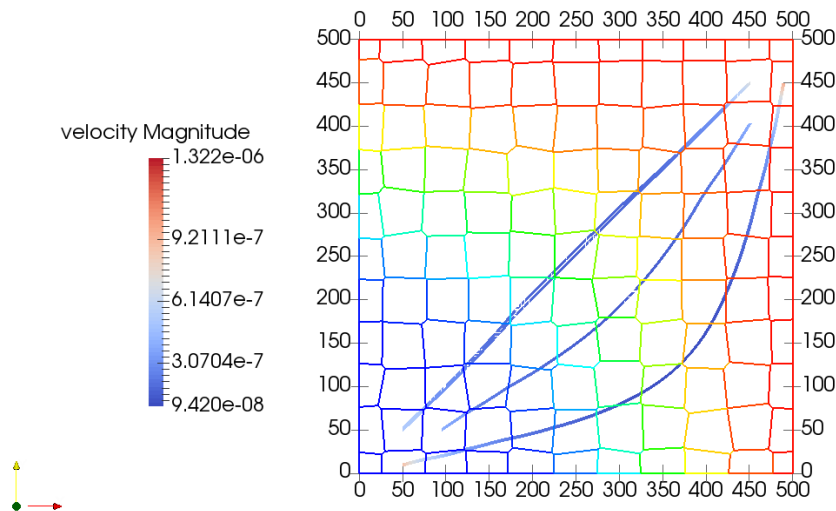


Figure 11 – Result visualization for the 2D_SemiRegular case.

4.4 2D_Voronoi

Results visualization are shown in Figure 12Figure 11.

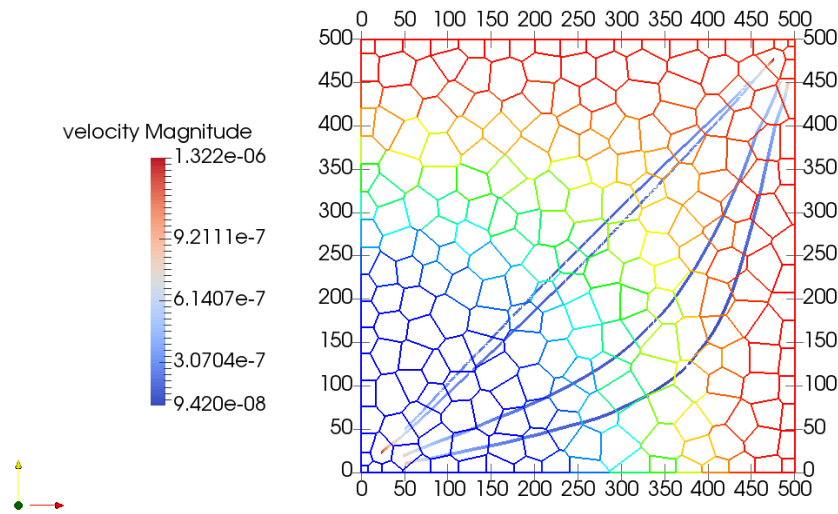


Figure 12 – Result visualization for the 2D_Voronoi case.

4.5 2D_CVT

Results visualization are shown in Figure 13

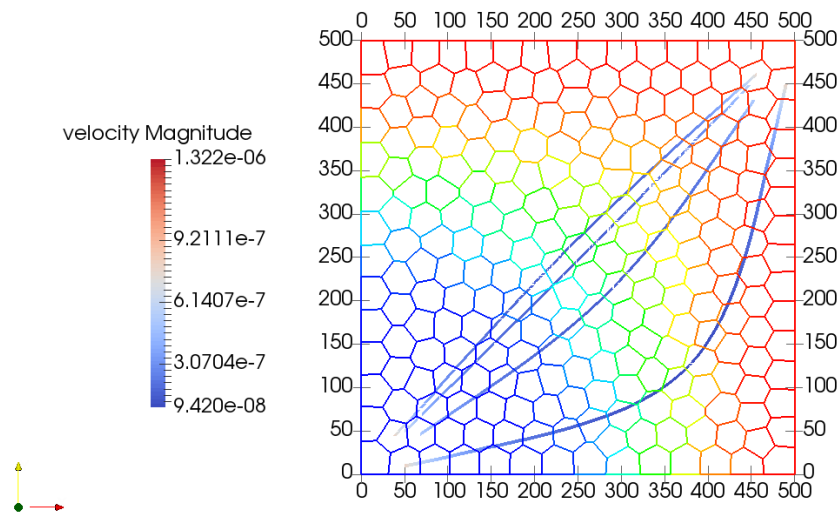
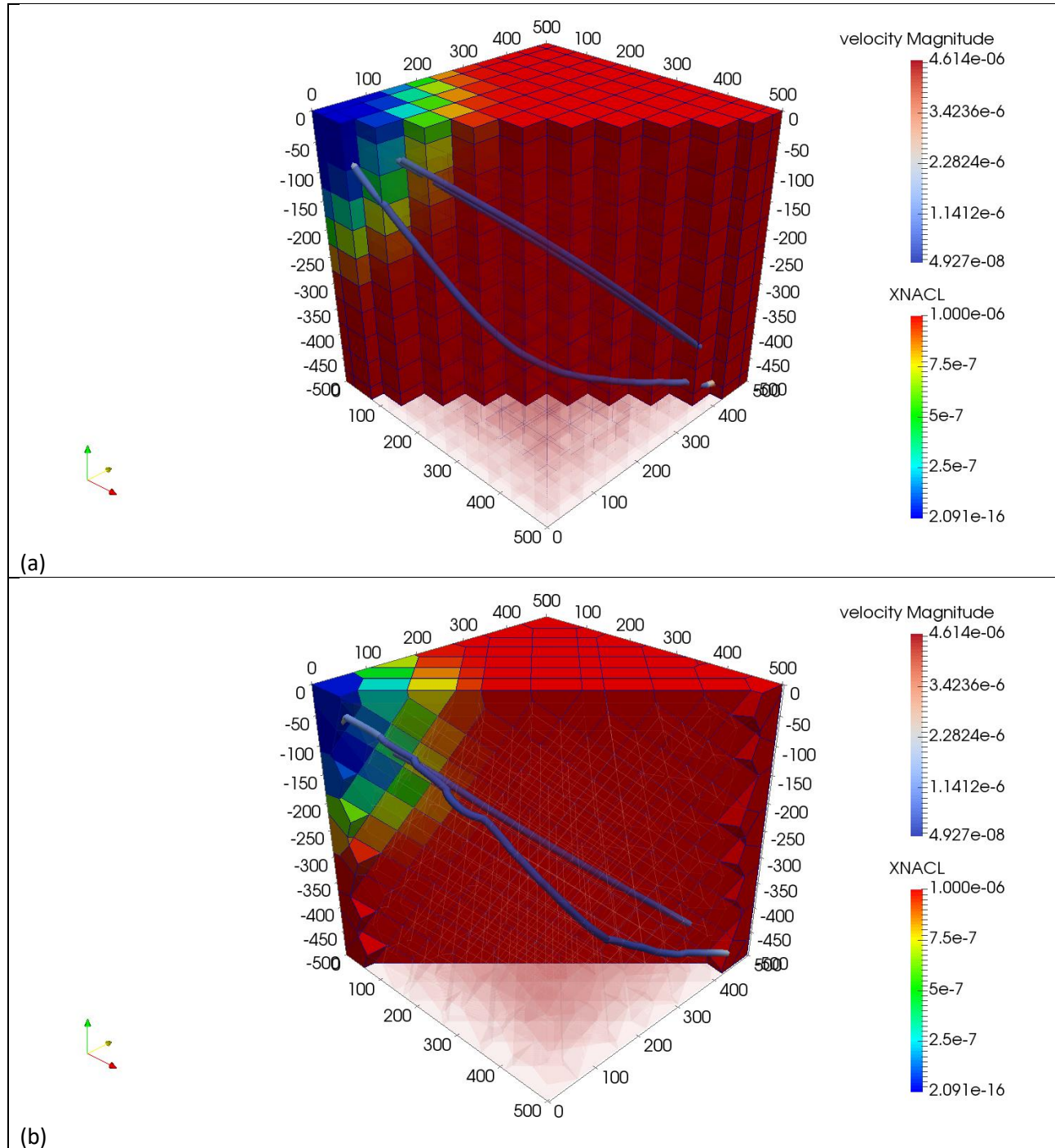


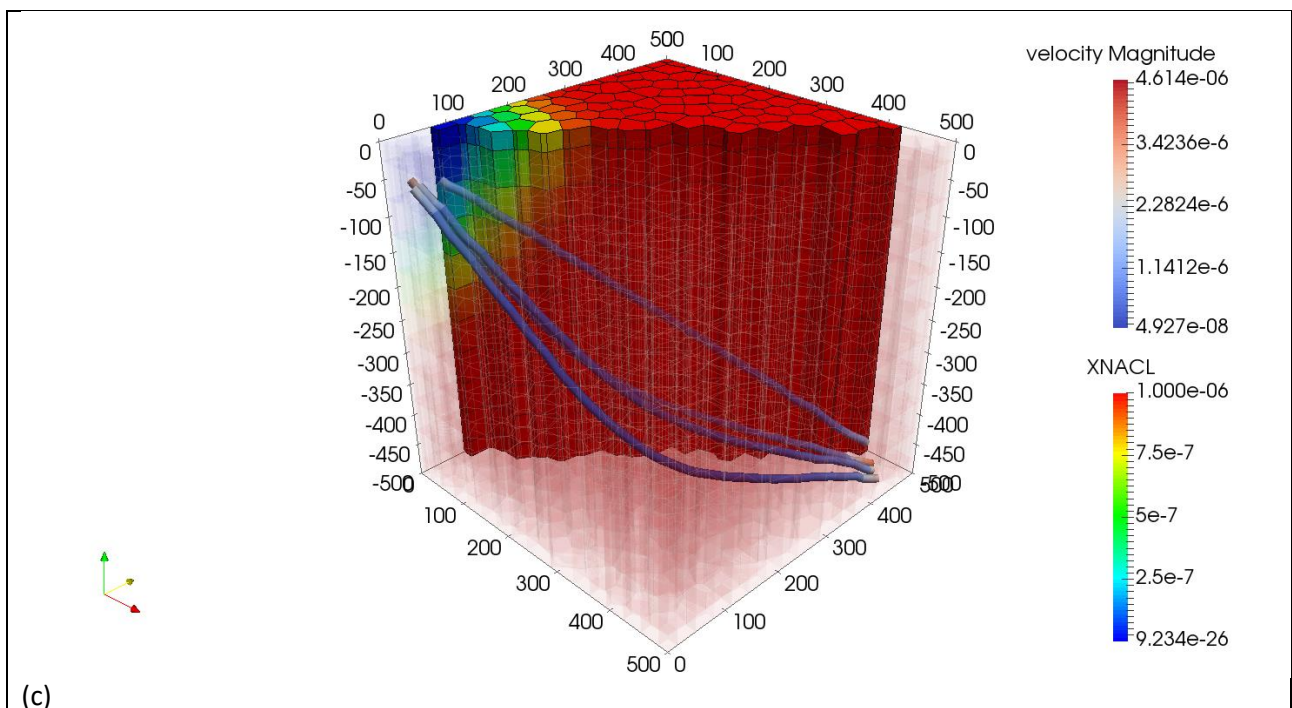
Figure 13– Result visualization for the 2D_CVT case.

4.6 3D Models

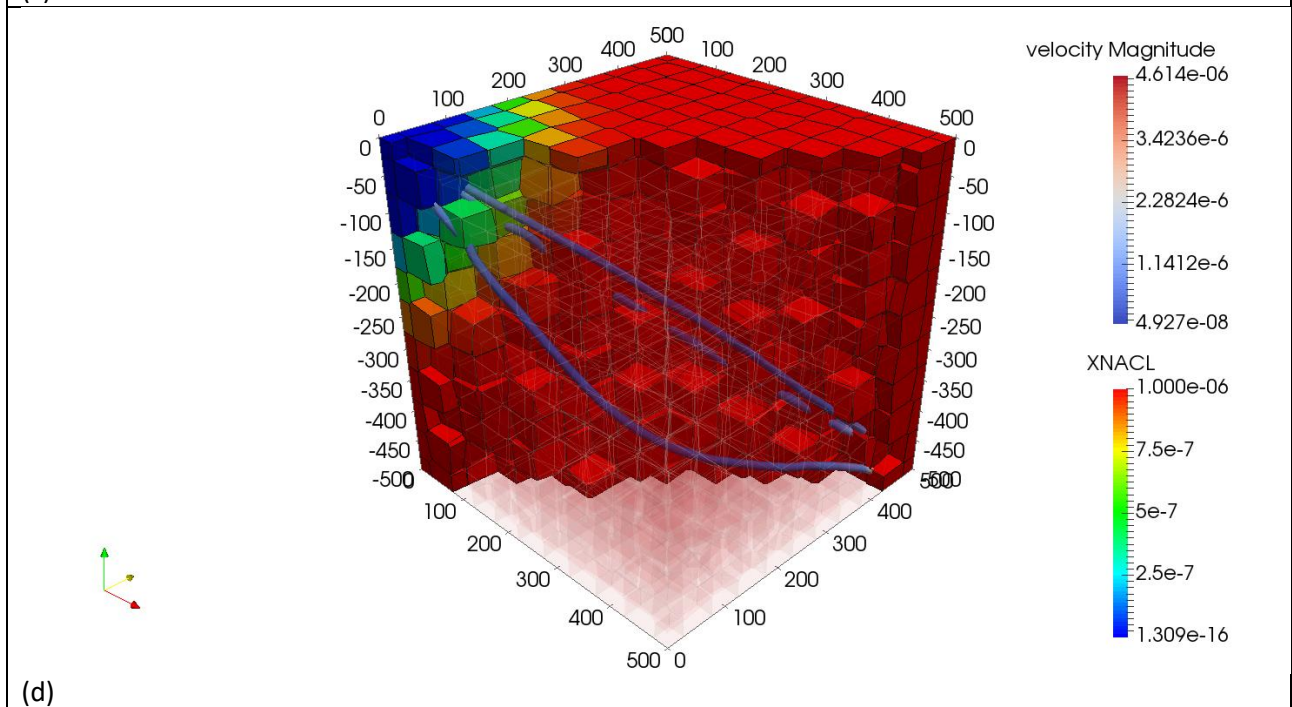
The 3D models refer to a simulation domain of a cube of $500 \times 500 \times 500 \text{ m}^3$. The simulation is isothermal and a injection of 1.0 kg/s is performed at $(0,0,500)$ and production of 1.0 kg/s at $(500,500,-500)$. The injection/production scenario is done using 6 grid type as show in Figure 14.

Similarly to the method of visualization of the 2D_Regular case explained at chapter 4.1.1, after calculating the pathline using TOUGH2Path is possible to obtain a graphic representation of the grid and of the pathlines as shown in Figure 14.





(c)



(d)

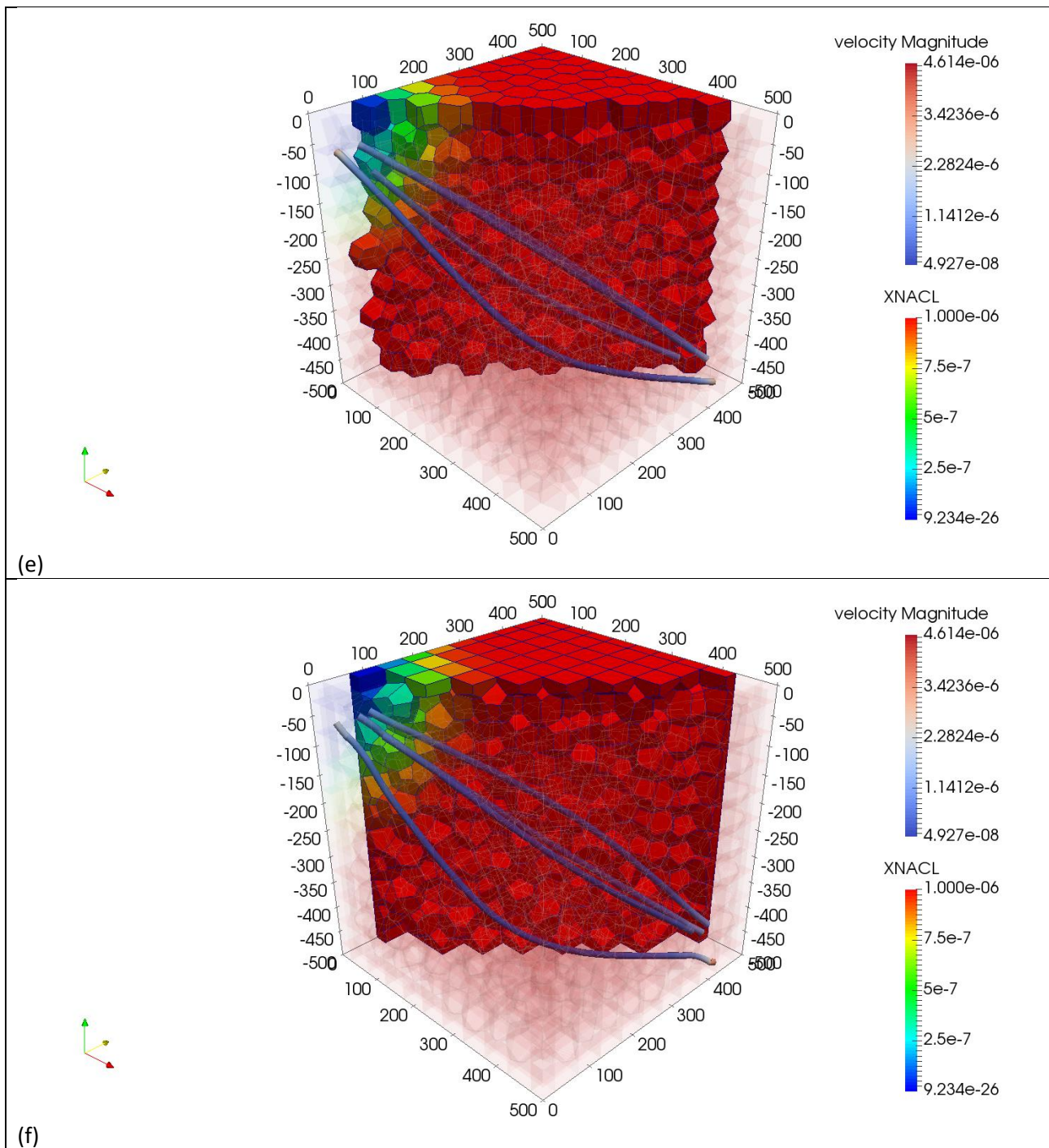


Figure 14 – 3D representation of the pathline for the 6 grid type: (a) 3D_Structured; (b) 3D_45_Structured; (c) 3D_Voronoi; (d) 3D_StructuredPerturbed; (e) 3D_CVT, (f) 3D_CVT2.

5 References

Pruess, K., Oldenburg, C., Moridis, G., 1999. TOUGH2 User's Guide, Version 2.0, Report LBNL-43134. Lawrence Berkeley National Laboratory, Berkeley, California, (Report LBNL-43134).

Klausen, R. A., Rasmussen, A. F., Stephansen, A. F., 2012. Velocity interpolation and streamline tracing on irregular geometries Comput. Geosci, 16:261–276.

Bonduá, S., Berry, P., Bortolotti, V., Cormio, C., 2012. TOUGH2Viewer: a postprocessing tool for interactive 3D visualization of locally refined unstructured grids for TOUGH2. Comput. Geosci. 46, 107–118.

Bonduà, S., A. Battistelli, P. Berry, V. Bortolotti, A. Consonni, C. Cormio, C. Geloni and E. M. Vasini 2017. 3D Voronoi grid dedicated software for modeling gas migration in deep layered sedimentary formations with TOUGH2-TMGAS, Computers & Geosciences 108(Supplement C), 50-55.