How to run the code (in windows only):

Prerequisites:

* Python 3

# pip install python3

* List of software in file requirements.txt

# pip install -r /path/to/requirements.txt

* Anaconda to run .ipynb files

https://www.anaconda.com/products/distribution

Development and analysis of structure is done through these steps, as visible in test.xml file:

1. Sphere aggregation <aggSph> to create spheres packing.
2. Voronoi tessellation <tess> starting from the centers of the spheres. Then edges and nodes of the tessellation are assessed; instead of edges and nodes, cylinders with diameter and spheres with diameter are inserted.
3. Morphological operations <morph> such as closing with a defined size .
4. Postprocessing <postProc> which includes postprocessing calculations.

Steps:

1. For the .xml files ending in ‘PAR’ in the ‘PlugIns’ folder: update paths based on your locations in the pc.
2. Edit the parameters of interest in test.xml file. Every plugin in the ‘PlugIns’ folder has within it a .xml file with the parameters used. There is one file ending with INI and another ending with PAR. The EXE and PAR files are then copied to a new folder with name and each parameter will be changed according to the choices in the test.xml file. In ‘launchPlugin.py’ there are all the operations to be done to launch a single function. First the folder is created, then the plugins are used.

In test.xml there are different items:

* : name of the output file. **Tip**: In the name, change the test number and related parameters so that it is easier to see the parameters from the folder name.
* : never change it, it links to ‘PlugIns’ folder.
* : number of spheres to be aggregated (not always reachable in case space runs out, so they will be placed in the same spot).
* and : more info are given in (Ferri et al., 2021). ‘They are used to control the compactness of the aggregates. The probability that the next object is positioned on a concave zone depends on a parameter called α and the probability that the concave zone is the closest to the center of mass of the aggregate depends on a β parameter (where β ≤ α).’
* : repulsion.
* : spheres diameter.
* : box size in integers.
* : spheres diameter to build the ‘skeleton’ for the tessellation.
* : cylinders diameter to build the ‘skeleton‘ for the tessellation. **Tip**: .
* : number of iterations for the Voronoi tessellation. **Tip:** 1 is good.
* : type of morphological operation. Generally closing.
* : voxel size for the chosen morphological operation.
* : voxel resolution.

Good parameters to start with:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Unit** | **Value** |
|  |  | 380 |
|  |  | 25 |
|  |  | 0.55 |
|  |  | 0.75 |
|  |  | 25 |
|  |  | 420 |
|  |  | 12 |
|  |  | 4 |
|  |  | closing |
|  |  | 1 |
|  |  | 18 |
|  |  | \* |

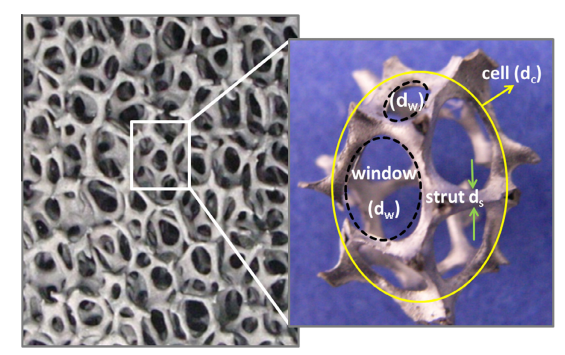
\* Depending on desired resolution

Tips:

* For some initial tests, you can fix some parameters and change others. For example, if you have the density of spheres within the volume, fix the number of spheres.
* If you need to increase / decrease the porosity , you can play with:
* with constant (but remind that the actual number of spheres can be lower than )
* , ,
* An analysis on Representative Elementary Volume () should be performed; that is defined as a volume around a point in which all averaged properties are independent on the size of the volume. (Bachmat & Bear, 1986) A box size of about may be good. (Agostini et al., 2022)
* If you have an idea of the pore size, the following equation can be used to evaluate the diameter of the cells / spheres for the initial aggregation (Inayat et al., 2012)

where is the window diameter in and is equivalent the pore size.

Then the voxel resolution is assessed as:



\*taken from (Inayat et al., 2012)

1. # python geoGenArgPostTort.py --name test.xml

Several postprocessing operations are performed, including:

* padding operation of the tiff file set (binarized, consisting of 0 and 1), via the numpy.pad function: layers of 0 are added to the edges (for simulation needs)
* exclusion of box edges, where a lot of material accumulates after the Voronoi tessellation (this should give a more realistic physical image)
* exclusion of the loose points via Blender (for simulation needs) – see file separate\_loose.py

if you want to analyze the pores, you can run the file pore\_size.ipynb. It must be manually added to the test folder. Anaconda console is required!

# jupyter notebook pore\_size.ipynb

1. Additionally, you can mesh through snappyHexMesh (generally two parameters are varied: number of cells in initial background grid and levels of refinement, no layers at the wall are needed). **Tip**: If you are dealing with small pore sizes, keep the mesh in the default units (after checking them in paraview) and then use this command to scale the size once the mesh is finished, where has been calculated previously

# transformPoints "scale=($ $ $)"

**References**:

Agostini, E., Boccardo, G., & Marchisio, D. (2022). An open-source workflow for open-cell foams modelling: Geometry generation and CFD simulations for momentum and mass transport. *Chemical Engineering Science*, *255*, 117583. https://doi.org/10.1016/j.ces.2022.117583

Bachmat, Y., & Bear, J. (1986). Macroscopic modelling of transport phenomena in porous media. 1: The continuum approach. *Transport in Porous Media*, *1*(3), 213–240. https://doi.org/10.1007/BF00238181

Ferri, G., Humbert, S., Digne, M., Schweitzer, J. M., & Moreaud, M. (2021). Simulation Of Large Aggregate Particles System With A New Morphological Model. In *Image Analysis and Stereology* (Vol. 40, Issue 2, pp. 71–84). https://doi.org/10.5566/ias.2488

Inayat, A., Klumpp, M., Lammermann, M., Freund, H., & Schwieger, W. (2012). Development of a new pressure drop correlation for open-cell foams based completely on theoretical grounds: Taking into account strut shape and geometric tortuosity. *Chemical Engineering Journal*, 704–719.