Manual for MetamaterialFinder: A Software Toolbox for Discovering Mechanical Metamaterials Based on Simple Closed Curves

# Overview

The software toolbox was built upon the ABAQUS scripting interface and implemented in Python [1]. The framework provides a text-based interface consisting of four individual json files: Meshes.json, Materials.json, Pores.json and Materials.json. Each file contains a collection of properties, effectively serving as databases. These files serve as input for three Python files, PoreFunctions.py, Materials.py and MetamaterialFinder.py. The file PoreFunctions.py contains a collection of pore functions. Each new function has to be added in this file. Materials.py provides classes for distinct material models. So far, only linear elastic material models are supported. New models can be added there. The third Python file gets executed by ABAQUS and collects the data from all the other files. A flow chart diagram of the toolbox is presented in Figure 1.

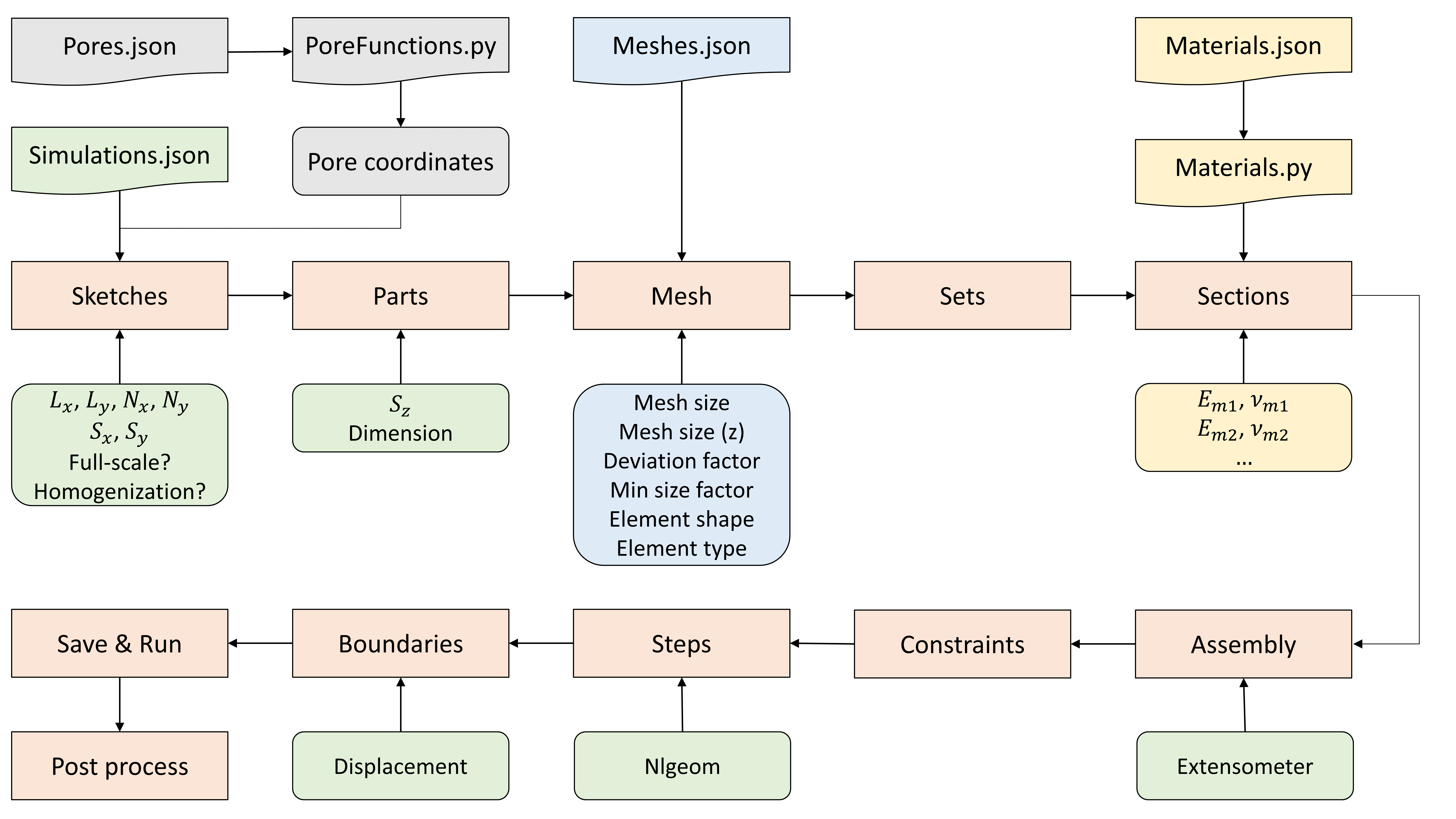


Figure : Flowchart diagram of the proposed framework. Red nodes indicate the individual steps performed for an individual simulation.

The paths to the ABAQUS plugins and additional files are specified in a configuration file, config.ini, as listed in Listing 1. Two ABAQUS plugins are needed for the full feature set of the toolbox: Find Nearest Node and Micromechanics. The first plugin is used to find the nearest nodes for the evaluation of extensometers. The second is needed to perform homogenization simulations.

|  |
| --- |
| **[ABAQUS]**  *path\_nearest\_node* **=** C:\SIMULIA\CAE\2019\win\_b64\code\python2.7\lib\abaqus\_plugins\findNearestNode  *path\_micromechanics* **=** C:\SIMULIA\CAE\plugins\2019\MicroMechanics  **[LOCAL]**  *path\_temp* **=** C:\temp  **[DATABASES]**  *materials\_json\_path* **=** Materials.json  *meshes\_json\_path* **=** Meshes.json  *pores\_json\_path* **=** Pores.json  *simulations\_json\_path* **=** Simulations.json |

Listing : Configuration file, config.ini, containing the paths to the ABAQUS plugins and database files.

In the following sections we present the format of the individual input files with a simple example structure: one radial pore in the middle of the unit cell made of one material (“Hard”), surrounded by another material (“Soft”) as depicted in Figure 2. Since the pore is fully contained in the unit cell, homogenization simulations are performed.

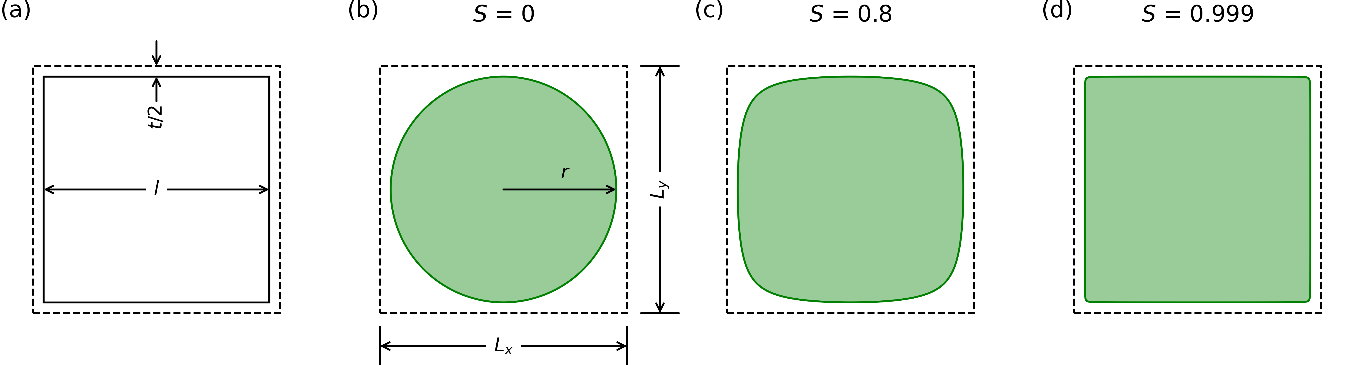


Figure : Example unit cell.

## Material Definition

Materials are defined in Materials.json. In the current state of the software toolbox only linear elastic materials are supported. Therefore, three material parameters are necessary to specify a material: Young’s modulus, Poisson’s Ratio and density. Two example materials (“Soft” and “Hard”) are listed in Listing 2. The descriptions and options of these parameters are listed in Table 1. The specifier (“LinearElasticMaterial” in this example) corresponds to a class of the same name defined in Materials.py. The names of the materials are used as references in Pores.json and Simulations.json. To add a new material model, a corresponding class needs to be added to Materials.py.

|  |
| --- |
| {  "LinearElasticMaterial": [{"name": "Soft", "E": 4, "nu": 0.35, "density": 1},  {"name": "Hard", "E": 1991, "nu": 0.35, "density": 1}]  } |

Listing : Material definition example (Materials.json) for two linear elastic materials.

Table : Material parameters as used in Materials.json.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Options** |
| Name | Name of the material | String, used as a reference in Pores.json and Simulations.json |
| E | Young’s modulus | Number |
| nu | Poisson’s Ratio | Number |
| density | Density | Number |

## Mesh Definition

The mesh parameters are defined in the file Meshes.json. An example of such a file is presented in Listing 3. The descriptions and possible values of the individual entries are listed in Table 2. The parameters “mesh\_size”, “mesh\_size\_z”, “deviation\_factor” and “min\_size\_factor” are used to control the size of the mesh. The parameter “mesh\_size\_z” is optional. It can be used to create bigger (or smaller) mesh elements in the z-direction. This can help to speed up simulations. If the parameter is not specified, the parameter “mesh\_size” is used instead. The parameters “element\_shape” and “element\_type” control the shape and type of the mesh elements, respectively. Different options, depending of the dimension of the simulation are available and listed in Table 2. The specifier (“3dfine” in the example) is used as a reference in Simulations.json.

|  |
| --- |
| {  "3dfine": {"mesh\_size": 0.3,  "mesh\_size\_z": 5,  "deviation\_factor": 0.1,  "min\_size\_factor": 0.1,  "element\_shape": "hex",  "element\_type": "quadratic"}  } |

Listing : Mesh definition example (Meshes.json) for 2D and 3D simulations.

Table : Parameters of the mesh as used in Meshes.json.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Options** |
| mesh\_size | Global element mesh size | Number |
| mesh\_size\_z | Element size in the z-direction | Number (optional parameter) |
| deviation\_factor | Measure of how much the element edges deviate from the original geometry | Number |
| min\_size\_factor | Curvature control | Number |
| element\_shape | Controls the shape of the mesh elements | 2D: quad, quad-dominated or tri  3D: hex, hex-dominated, tet or wedge |
| element\_type | Controls the type of the mesh elements | 2D: linear-plane-stress, quadratic-plane-stress, linear-plane-strain or quadratic-plane-strain  3D: linear or quadratic |

## Pore Definition

The pores are defined in Pores.json. Parameter studies and their respective values are mainly defined in this file. An example is listed in Listing 4 for a single pore located at the center of the unit cell, filled with material “Hard” (as specified in Materials.json). The parameter “Pore functions” is defined by a list of pore function names corresponding to functions with the same name as defined in PoreFunctions.py. The number of pores is defined here. For each defined pore, a corresponding set of parameters have to be defined in “Pore function parameters”, “Pore displacements”, “Pore angles” and “Pore materials”. The parameter “Pore function parameters” is defined by a list of pore function ranges with general form “” where is the start value of the -th parameter of the function, is the end value of the -th parameter of the function and is the amount of steps created between the start and end value. ”. If more than one function parameter is defined with multiple values, all permutations are generated. Since a circle has only one geometric parameter, this reduces to “”. In the example (Listing 4) 91 values between 1 mm and 10 mm are created. This corresponds to a step size of 0.1 mm. The pore displacements are defined as a list of dictionaries with x- and y-coordinates. Here, also the variables of the unit cell, and , can be used for relative positioning of the pores. Similar to the pore function parameters, the pore positions and pore angles can be defined as a parameter study. The pore materials can either be “empty”, resulting in a cut out of the pore, or a name as defined in Materials.json. To avoid non-valid geometries, the parameters can be restricted by a set of relations which have to be fulfilled. For example, the relations “P1X1<LX/2.” and “P1X1<LY/2.” as used in the example, restricts the first parameter (X1) of the first pore (P1) to values lower than half the unit cell side lengths and . This ensures that the circle does not extend the unit cell. In general, (Pn) and (Xm) refer to the -th geometric parameter of the -th pore. The descriptions and possible values of the individual entries are listed in Table 3.

|  |
| --- |
| {  "CircleCenter": {"Pore functions": ["circle"],  "Pore function parameters": ["1;10;91"],  "Pore displacements": [{"X": "LX/2.;LX/2.;1",  "Y": "LY/2.;LY/2.;1"}],  "Pore angles": ["0;0;1"],  "Pore materials": ["Hard"],  "Pore relations": ["P1X1<LX/2.", “P1X1<LY/2.”]}  } |

Listing : Example pore definition (Pores.json) for a parameter study with a single pore located at the center of the unit cell filled with material “Hard”.

Table : Pore function parameters as defined in Pores.json.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Options** |
| Pore functions | List of pore functions | Name, must be present in PoreFunctions.py |
| Pore function parameters | List of pore function parameters | General form: |
| Pore displacements | List of pore displacements | General form: |
| Pore angles | List of pore angles | General form: |
| Pore materials | List of pore materials | Name, must be present in Materials.json |
| Pore relations | List of geometric restrictions | Inequalities with references to and to restrict the possible ranges of geometric parameters |

## Simulation Definition

Additional simulation details are defined in Simulations.json. An example is listed in Listing 5. The corresponding descriptions of these parameters are listed in Table 4. The keyword “Active” can be used to indicate whether or not a simulation should be executed. The unit cell dimension are specified as “||”. If a full-scale simulation is requested, the amount of unit cells in x- and y-direction are specified as “|” with total dimensions as “|”. The parameter “Outside material” and “Mesh” must be specified as names defined in Materials.json and Meshes.json, respectively. The “Dimension” can either be “3D” or “2D”, depending on the type of simulation. The “Pores” refer to a name defined in Pores.json. “Force full” and “Force homogenization” can be used to either perform a full-scale or homogenization simulation, respectively. “|” specifies the displacement of the reference point in x- and y-direction in percent of and , respectively. By setting “Nlgeom” to 1, non-linear geometry deformations are considered. When performing full-scale simulations, the parameter “Stepsize” defines the size between steps, where a static analysis is performed. After a simulation is finished, all output files are copied to the specified “Save directory”.

|  |
| --- |
| {  "CircleCenterUC": {  "Active": 1,  "LX|LY|LZ": "15|15|0.2",  "NX|NY": "1|1",  "SX|SY": "15|15",  "Outside material": "Soft",  "Mesh": "3dfine",  "Pores": "CircleCenter",  "Force full": 0,  "Force homogenization": 1,  "DX|DY": "0|7",  "Nlgeom": 0,  "Stepsize": 0.02,  "Dimension": "3D",  "Extensometer": {"P1": ["1.5\*LX", "LY/2"],  "P2": ["2\*LX+LX/2", "1.5\*LY"],  "P3": ["1.5\*LX", "2\*LY+LY/2"],  "P4": ["LX/2", "1.5\*LY"]},  "Save directory": "C:\\temp\\CIRCLE"}  } |

Listing : Example simulation definition (Simulations.json) for a homogenization simulation.

Table : Simulation parameters as defined in Simulations.json.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Options** |
| Active | Simulations are only performed if this parameter is set to 1 | 0, 1 (Boolean) |
| LX|LY|LZ | Bounding box dimensions in x-, y- and z-direction of the unit cell | Number|Number|Number |
| NX|NY | Amount of unit cell copies in x- and y-direction | Number|Number |
| SX|SY | Side length of the full-size structure in x and y | Number|Number |
| Outside material | Material to be used outside the pores | Must be present in Materials.json |
| Mesh | Mesh to be used for the simulation | Must be present in Meshes.json |
| Pores | Pores to be used for the simulation | Must be present in Pores.json |
| Force full | Creates a full-scale simulation, even if a homogenization is possible | 0, 1 (Boolean) |
| Force homogenization | Creates a homogenization simulation | 0, 1 (Boolean) |
| DX|DY | Displacement in x- and y-direction of the reference point | Number|Number |
| Nlgeom | Indicates if non-linear geometry should be used | 0, 1 (Boolean) |
| Dimension | Dimension of the simulation | Either “2D” or “3D” |
| Stepsize | Size between steps | Number |
| Extensometer | List of coordinates at which the mesh positions should be recorded | Dictionary |

## Example Results

The mechanical properties obtained from the example discussed in the previous section are depicted in Figure 3.

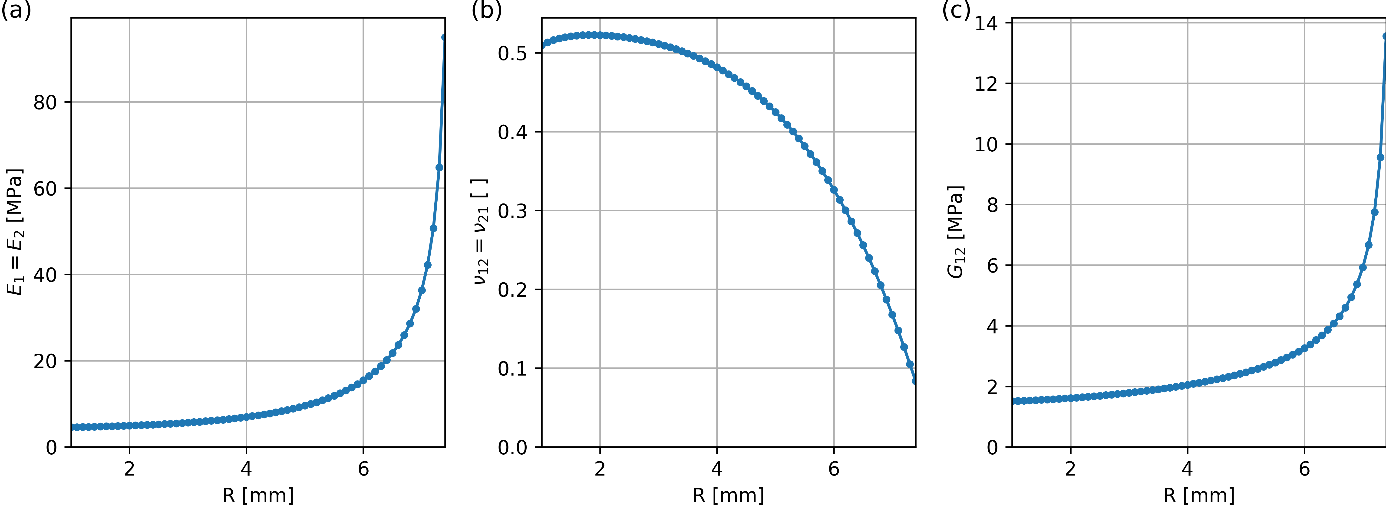


Figure : Mechanical properties obtained from homogenization simulations for the example structure as a function of the circular pore radius. (a): Young’s Modulus, (b) Poisson’s Ratio and (c) shear modulus.

# Input Files for Additional Examples

## Squircle

### Materials.json

|  |
| --- |
| {  "LinearElasticMaterial": [{"name": "Soft", "E": 4, "nu": 0.35, "density": 1},  {"name": "Hard", "E": 1991, "nu": 0.35, "density": 1}]  } |

### Meshes.json

|  |
| --- |
| {  "3dfine": {"mesh\_size": 0.3,  "mesh\_size\_z": 5,  "deviation\_factor": 0.1,  "min\_size\_factor": 0.1,  "element\_shape": "hex",  "element\_type": "quadratic"}  } |

### Pores.json

|  |
| --- |
| {  "SquircleCenter": {"Pore functions": ["squircle"],  "Pore function parameters": ["6.85;6.85;1|0;0.999;21"],  "Pore displacements": [{"X": "LX/2.;LX/2.;1",  "Y": "LY/2.;LY/2.;1"}],  "Pore angles": ["0;0;1"],  "Pore materials": ["empty"],  "Pore relations": ["P1X1<LX/2.", "P1X1<LY/2."]}  } |

### Simulations.json

|  |
| --- |
| {  "SquircleCenterUC": {  "Active": 1,  "LX|LY|LZ": "15|15|0.2",  "NX|NY": "1|1",  "SX|SY": "15|15",  "Outside material": "Hard",  "Mesh": "3dfine",  "Pores": "SquircleCenter",  "Force full": 0,  "Force homogenization": 1,  "DX|DY": "0|7",  "Nlgeom": 0,  "Stepsize": 0.02,  "Dimension": "3D",  "Extensometer": {"P1": ["1.5\*LX", "LY/2"],  "P2": ["2\*LX+LX/2", "1.5\*LY"],  "P3": ["1.5\*LX", "2\*LY+LY/2"],  "P4": ["LX/2", "1.5\*LY"]},  "Save directory": "C:\\temp\\SQUIRCLE"}  } |

## Tetra-Antichiral (Piecewise)

### Materials.json

|  |
| --- |
| {  "LinearElasticMaterial": [{"name": "Soft", "E": 4, "nu": 0.35, "density": 1},  {"name": "Hard", "E": 1991, "nu": 0.35, "density": 1}]  } |

### Meshes.json

|  |
| --- |
| {  "3dfine": {"mesh\_size": 0.3,  "mesh\_size\_z": 5,  "deviation\_factor": 0.1,  "min\_size\_factor": 0.1,  "element\_shape": "hex",  "element\_type": "quadratic"}  } |

### Pores.json

|  |
| --- |
| {  "TetraAntichiral": {"Pore functions": ["antichiral"],  "Pore function parameters": ["5.0;5.0;1|25.0;25.0;0|0.5;1.5;21"],  "Pore displacements": [{"X": "LX/2.;LX/2.;1",  "Y": "LY/2.;LY/2.;1"},  {"X": "0.;0.;1",  "Y": "LY/2.;LY/2.;1"},  {"X": "LX;LX;1",  "Y": "LY/2.;LY/2.;1"},  {"X": "0.;0.;1",  "Y": "LY;LY;1"},  {"X": "LX/2.;LX/2.;1",  "Y": "LY;LY;1"},  {"X": "LX;LX;1",  "Y": "LY;LY;1"},  {"X": "0.;0.;1",  "Y": "0.;0.;1"},  {"X": "LX/2.;LX/2.;1",  "Y": "0.;0.;1"},  {"X": "LX;LX;1",  "Y": "0.;0.;1"}],  "Pore angles": ["90;90;1", "0;0;1", "0;0;1", "90;90;1", "0;0;1", "90;90;1", "90;90;1", "0;0;1", "90;90;1"],  "Pore materials": ["empty", "empty", "empty", "empty", "empty", "empty", "empty", "empty", "empty"]}  } |

### Simulations.json

|  |
| --- |
| {  "TetraAntichiralUC": {  "Active": 1,  "LX|LY|LZ": "50|50|0.2",  "NX|NY": "1|1",  "SX|SY": "50|50",  "Outside material": "Soft",  "Mesh": "3dfine",  "Pores": "TetraAntichiral",  "Force full": 0,  "Force homogenization": 1,  "DX|DY": "0|7",  "Nlgeom": 0,  "Stepsize": 0.02,  "Dimension": "3D",  "Extensometer": {"P1": ["1.5\*LX", "LY/2"],  "P2": ["2\*LX+LX/2", "1.5\*LY"],  "P3": ["1.5\*LX", "2\*LY+LY/2"],  "P4": ["LX/2", "1.5\*LY"]},  "Save directory": "C:\\temp\\ANTICHIRAL"}  } |

## Tetra-Antichiral (Sphylinder)

### Materials.json

|  |
| --- |
| {  "LinearElasticMaterial": [{"name": "Soft", "E": 4, "nu": 0.35, "density": 1},  {"name": "Hard", "E": 1991, "nu": 0.35, "density": 1}]  } |

### Meshes.json

|  |
| --- |
| {  "3dfine": {"mesh\_size": 0.3,  "mesh\_size\_z": 5,  "deviation\_factor": 0.1,  "min\_size\_factor": 0.1,  "element\_shape": "hex",  "element\_type": "quadratic"}  } |

### Pores.json

|  |
| --- |
| {  "TAC-Sphylinder": {"Pore functions": ["sphylinder\_projection"],  "Pore function parameters": ["15.0;15.0;1|7.5;7.5;1|0.0;0.999;21"],  "Pore displacements": [{"X": "LX/2.;LX/2.;1",  "Y": "LY/2.;LY/2.;1"},  {"X": "0.;0.;1",  "Y": "LY/2.;LY/2.;1"},  {"X": "LX;LX;1",  "Y": "LY/2.;LY/2.;1"},  {"X": "0.;0.;1",  "Y": "LY;LY;1"},  {"X": "LX/2.;LX/2.;1",  "Y": "LY;LY;1"},  {"X": "LX;LX;1",  "Y": "LY;LY;1"},  {"X": "0.;0.;1",  "Y": "0.;0.;1"},  {"X": "LX/2.;LX/2.;1",  "Y": "0.;0.;1"},  {"X": "LX;LX;1",  "Y": "0.;0.;1"}],  "Pore angles": ["90;90;1", "0;0;1", "0;0;1", "90;90;1", "0;0;1", "90;90;1", "90;90;1", "0;0;1", "90;90;1"],  "Pore materials": ["empty", "empty", "empty", "empty", "empty", "empty", "empty", "empty", "empty"]}  } |

### Simulations.json

|  |
| --- |
| {  "TetraAntichiralSphylinderUC": {  "Active": 1,  "LX|LY|LZ": "50|50|0.2",  "NX|NY": "1|1",  "SX|SY": "50|50",  "Outside material": "Soft",  "Mesh": "3dfine",  "Pores": "TAC-Sphylinder",  "Force full": 0,  "Force homogenization": 1,  "DX|DY": "0|7",  "Nlgeom": 0,  "Stepsize": 0.02,  "Dimension": "3D",  "Extensometer": {"P1": ["1.5\*LX", "LY/2"],  "P2": ["2\*LX+LX/2", "1.5\*LY"],  "P3": ["1.5\*LX", "2\*LY+LY/2"],  "P4": ["LX/2", "1.5\*LY"]},  "Save directory": "C:\\temp\\SPHYLINDER"}  } |

## Hippopede

### Materials.json

|  |
| --- |
| {  "LinearElasticMaterial": [{"name": "Soft", "E": 4, "nu": 0.35, "density": 1},  {"name": "Hard", "E": 1991, "nu": 0.35, "density": 1}]  } |

### Meshes.json

|  |
| --- |
| {  "2dfine-plane-strain": {"mesh\_size": 0.5,  "deviation\_factor": 0.1,  "min\_size\_factor": 0.1,  "element\_shape": "quad",  "element\_type": "quadratic-plane-strain"}  } |

### Pores.json

|  |
| --- |
| {  "Hippopede": {"Pore functions": ["hippopede"],  "Pore function parameters": ["1;5;21|1;5;21"],  "Pore displacements": [{"X": "LX/4.;LX/4.;1",  "Y": "LY/4.;LY/4.;1"},  {"X": "3\*LX/4.;3\*LX/4.;1",  "Y": "LY/4.;LY/4.;1"},  {"X": "3\*LX/4.;3\*LX/4.;1",  "Y": "3\*LY/4.;3\*LY/4.;1"},  {"X": "LX/4.;LX/4.;1",  "Y": "3\*LY/4.;3\*LY/4.;1"}],  "Pore angles": ["0;0;1", "90;90;1", "0;0;1", "90;90;1"],  "Pore materials": ["empty"],  "Pore relations": ["P1X1>P1X2", "2\*P1X1+P1X2<LX/2.-0.2"]}  } |

### Simulations.json

|  |
| --- |
| {  "HippopedeFull": {  "Active": 1,  "LX|LY|LZ": "20|20|0.2",  "NX|NY": "4|4",  "SX|SY": "85|90",  "Outside material": "Hard",  "Mesh": "2dfine-plane-strain",  "Pores": "Hippopede",  "Force full": 1,  "Force homogenization": 0,  "DX|DY": "0|2",  "Nlgeom": 0,  "Stepsize": 0.04,  "Dimension": "2D",  "Extensometer": {"P1": ["1.5\*LX", "1.5\*LY"],  "P2": ["2.5\*LX", "1.5\*LY"],  "P3": ["2.5\*LX", "2.5\*LY"],  "P4": ["1.5\*LX", "2.5\*LY"]},  "Save directory": **"C:\\temp\\HIPPOPEDE"**}  } |

## Multi-Material Sphylinder

### Materials.json

|  |
| --- |
| {  "LinearElasticMaterial": [{"name": "Soft", "E": 4, "nu": 0.35, "density": 1},  {"name": "Hard", "E": 1991, "nu": 0.35, "density": 1}]  } |

### Meshes.json

|  |
| --- |
| {  "2dfine-plane-stress": {"mesh\_size": 0.5,  "deviation\_factor": 0.1,  "min\_size\_factor": 0.1,  "element\_shape": "quad",  "element\_type": "quadratic-plane-stress"}  } |

### Pores.json

|  |
| --- |
| {  "FourSphylinders": {"Pore functions": ["sphylinder\_projection"],  "Pore function parameters": ["1.0;7.0;16|1.0;7.0;16|0.9;0.9;1"],  "Pore displacements": [{"X": "LX/4.;LX/4.;1",  "Y": "LY/4.;LY/4.;1"},  {"X": "3\*LX/4.;3\*LX/4.;1",  "Y": "LY/4.;LY/4.;1"},  {"X": "3\*LX/4.;3\*LX/4.;1",  "Y": "3\*LY/4.;3\*LY/4.;1"},  {"X": "LX/4.;LX/4.;1",  "Y": "3\*LY/4.;3\*LY/4.;1"}],  "Pore angles": ["0;0;1", "90;90;1", "0;0;1", "90;90;1"],  "Pore materials": ["Hard"],  "Pore relations": ["P1X1+P1X2<LX/2.", "P1X1+P1X2<LY/2."]}  } |

### Simulations.json

|  |
| --- |
| {  "SphylinderMulti": {  "Active": 1,  "LX|LY|LZ": "20|20|0.2",  "NX|NY": "3|3",  "SX|SY": "65|70",  "Outside material": "Soft",  "Mesh": "2dfine-plane-stress",  "Pores": "FourSphylinders",  "Force full": 1,  "Force homogenization": 0,  "DX|DY": "0|5",  "Nlgeom": 1,  "Stepsize": 0.04,  "Dimension": "2D",  "Extensometer": {"P1": ["1.5\*LX", "LY"],  "P2": ["2\*LX", "1.5\*LY"],  "P3": ["1.5\*LX", "2\*LY"],  "P4": ["LX", "1.5\*LY"]},  "Save directory": **"C:\\temp\\SPHYLINDERMULTI"**}  } |

References

[1] G. van Rossum, F.L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, 2009.