MOAMMM

Multi-scale Optimisation for Additive Manufacturing of fatigue resistant shock-absorbing MetaMaterials — MOAMMM

Tutorial for lattice generation — Voxelisation and Finite Element Discretisation

L. Cobian (IMDEA Materials), A. Hössinger-Kalteis (JKU), L. Wu (ULiege), L. Noels (ULiege)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862015

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862015

1. General information

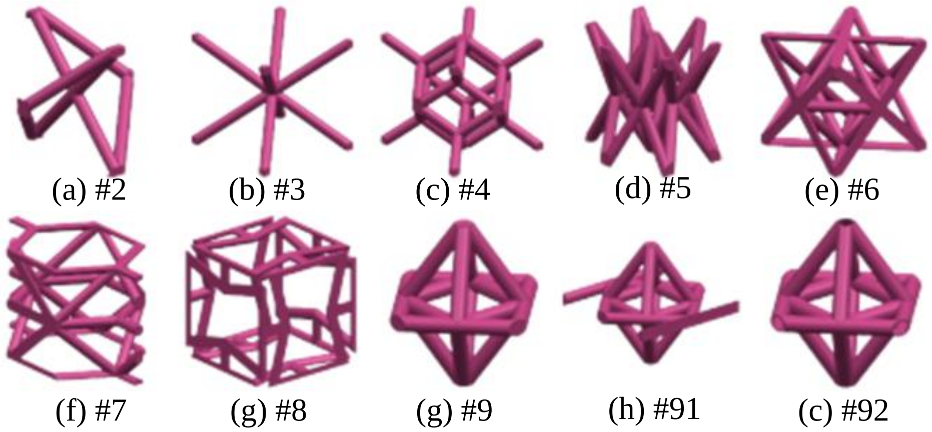


Figure 1: Lattice structures collection [FRE17] [LEH18].

This deliverable provides code to generate volume elements of the lattice structures illustrated in Figure 1. Two options exist

- Generate voxelized and/or finite element discretization of a Representative Volume Element (RVE) of given cells with given geometrical properties;
- Generate Stochastic Volume Elements with random geometrical properties.

The code is available from the project repository: https://gitlab.uliege.be/moammm/moammmpublic/-/tree/master/code/Lattice_Cell_script.

Links have been made from the project web-site: https://www.moammm.eu/index.php/developed-code/

2. Compilation of the code

You need

- gfortran compiler
- python3
- f2py3 package
- scipy for python3

Run the script ./compile.sh or manually generate the files (with your selected version of python, here python3)

- f2py3 -c vmf_bisection.f95 -m vmf_bisection
- f2py3 -c vmf_hollow_ID.f95 -m vmf_hollow_ID
- f2py3 -c vmf_hollow_OD.f95 -m vmf_hollow_OD
- f2py3 -c vmf_test.f95 -m vmf_test

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862015

3. Generation of RVEs of given geometrical properties

Run program "python3 geometry_parametric.py" and step by step introduce the parameters as indicated.

1. Choose geometry by choosing number (2,3,4,5,6,7,8,9,91 or 92), see Figure 1.
2. Indicate no. voxels per side of "unit cell box" (fixed by default in 128).
3. Program will ask if you want to introduce:
   - volume fraction and it calculates the radius (not valid for geo 92)
   - radius and it calculates the volume fraction
4. Introduce number of unit cells wanted in each direction:
   - length: no. unit cells in x direction
   - width: no. unit cells in y direction
   - height: no. unit cells in z direction
5. Introduce size of unit cell "box". Adimensional. Value 1 recommended
   - For geo 7: You only introduce dimension of y and z direction. It automatically calculates x direction size.
   - If you chose a) in step 3. go to 6.a
   - If you chose b) in step 3. go to 6.b
6. a. Introduce desired volume fraction that occupies the struts in unit cell size. Value valid from 0 to 1 (0.5 is 50%).
   - It calculates the radius value, printed as RADIUSf.
   - Outputs: vox_nvoxels.geo and phase_map_nvoxels.geo
   - It produces the .geo file for meshing the geometry in gmsh (https://gmsh.info/).
   - Output: ngeo_nvoxels.geo
6. b. Introduce desired struts radio value (adimensional, same magnitude as cell size)
   - For geo 92: outer radius.
   - It calculates the volume fraction value, printed as densityf.
   - Outputs: vox_nvoxels.geo, phase_map_nvoxels.geo
   - It produces the .geo file for meshing the geometry in gmsh (https://gmsh.info/).
   - Output: ngeo_nvoxels.geo

Other modifiable parameters for certain geometries:

1. Geo 8: Auxetic bow-tie, see Figure 2.
   - bow_h (adimensional, same magnitude as size of unit cell, should be lower than size of unit cell)
   - bow_angle (degrees°, is limited by $atan\left(\frac{size\ of\ unit\ cell}{bow\_h}\right)\frac{180}{\pi}$)
2. Geo 92: hollow variant of Geo 9
   - r= outer radius
   - ri= inner radius
   - by default: ri= r/1.25 (hard-coded, line 506 of Auto_geometry_parametric.py)
   - To calculate volume fraction: calculate volume fractions of geo 9 with radius values of inner and outer radius and obtain difference.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862015

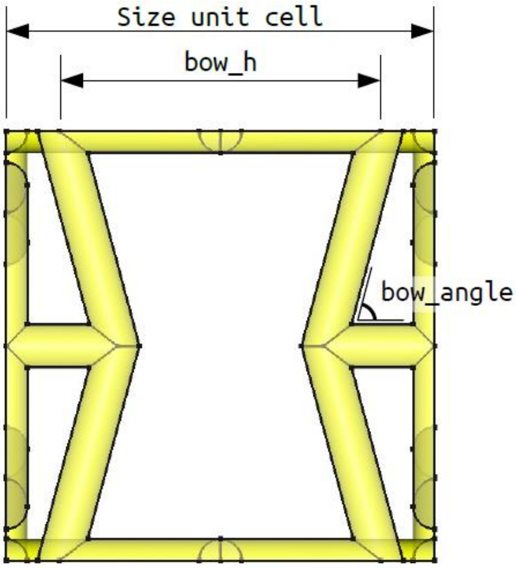


Figure 2: Geo 8 - Auxetic bow-tie.

4. Generation of Stochastic Volume Elements (SVEs)

The program “python3 MicroSample.py” adds a layer to generate random SVE. The following lines can be changed:

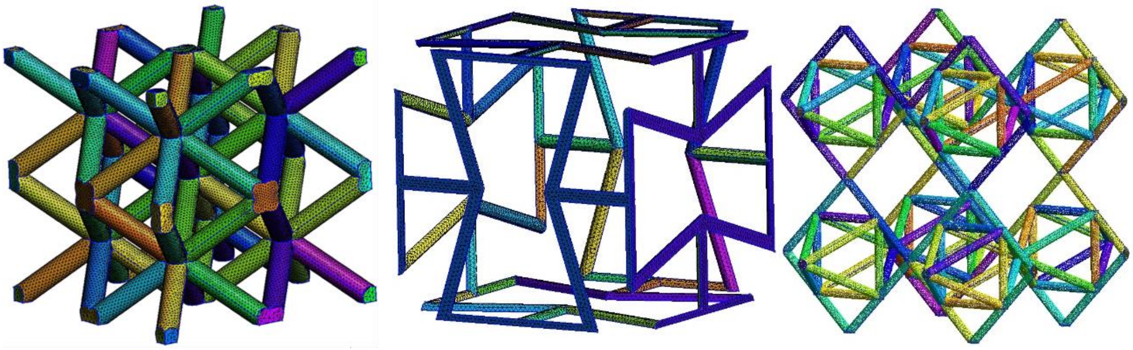
- cellType = "3": cell type from "2", "3", "4", "5", "6", "7", "8", "9", "91" (or "92" if typeRand ="b"), see Figure 1.
- typeRand = "a" : "a" for random volume fraction and "b" for random radius
- nx =2: number of cell along x
- ny =2: number of cell along y
- nz =2: number of cell along z
- VmFrMin = 0.05: lower bound of random volume fraction, for geometry 8: 0.15 is a maximum, for other geometry 1. is a maximum
- VmFrMax = 0.25: max bound of random volume fraction, for geometry 8: 0.15 is a maximum, for other geometry 1. is a maximum
- sizemin=0.2: minimum size of unit cell
- sizemax=2.5: maximum size of unit cell
- radiusmin =0.01: minimum ratio of radius with respect to cell length
- radiusmax =0.1: maximum ratio of radius with respect to cell length
- bow_h_min =0.5: minimum bow of cell 8 with respect to cell length, see Figure 2

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 862015

- bow_h_max = 0.8: maximum bow of cell 8 with respect to cell length, see Figure 2
- bow_angle_min=65: minimum bow angle of cell 8, automatically corrected from bow_h if too low, see Figure 2
- bow_angle_max=85: maximum bow angle of cell 8, should be lower than 90, see Figure 2

Random values are taken between the provided bounds.

5. Examples

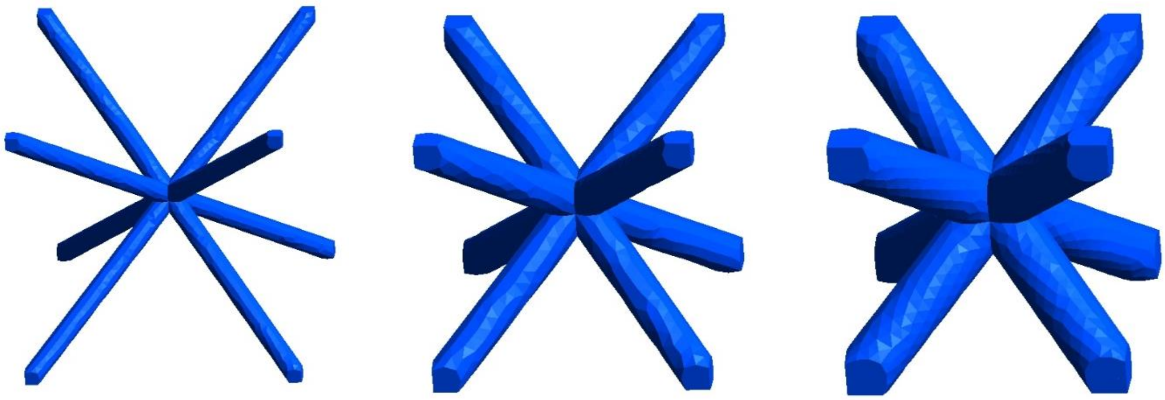


(a) $2 \times 2 \times 2$ cells #3
$V_f = 0.1402$; r=0.0202

(b) $1 \times 1 \times 1$ cell #8
$V_f = 0.0976$; r=0.05

(c) $2 \times 2 \times 2$ cells #9
$V_f = 0.214$; r=0.023

Figure 3: Different arrays of different cells of volume fraction V_f and struts radius r.



(a) $V_f = 0.046$; l=1.70

(b) $V_f = 0.095$; l=1.72

(c) $V_f = 0.18$; l=1.50

Figure 4: Different cell #3 of volume fraction V_f and length l.

Figure 3 illustrates realisations of three arrays of different cell-types, while Figure 4 illustrates three different realisations of the same cell number 3.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862015

6. Material model

The finite-strain visco-elastic-visco-plastic model is implemented as a cross platform UMAT subroutine that is openly available on https://gitlab.uliege.be/moammm/moammmpublic/-/tree/master/code/MaterialModels/FiniteStrain/Finite_VEVP (Licensed under the Creative Commons Attribution 4.0 International (CC BY 4.0) (https://creativecommons.org/licenses/by/4.0).) and referenced on the project web-site https://www.moammm.eu/index.php/developed-code/.

The model is also implemented at the cm3 laboratory, who provides efficient algorithms for homogenisation implemented in a parallel setting in the open source finite element code base on Gmsh and distributed under the GNU General Public License (GPL),. In order to have access to the sources (with no guarantee on the support):

- Register on https://gitlab.onelab.info/gmsh/gmsh.git
- Request code access to Ludovic Noels (L.Noels @ ulg.ac.be)
- Get access to code
    - git clone https://gitlab.onelab.info/gmsh/gmsh.git
    - git clone https://gitlab.onelab.info/cm3/cm3Libraries.git
    - Follow instruction in cm3Libraries/cm3apps/install.txt

Material parameters have been identified from macro-bulk samples (and are not fully representative of the struts behaviour):

- All the data are available on https://gitlab.uliege.be/moammm/moammmpublic/-/tree/master/ModelIdentification/deterministicIndentificationMacroBulkPA12RT and on http://dx.doi.org/10.5281/zenodo.6903647.
- Links have been made from the project web-site: https://www.moammm.eu/index.php/developed-code/ and https://www.moammm.eu/index.php/open-data/

7. References

[FRE17] T. Frenzel, K. Muamer, M. Wegener, Three-dimensional mechanical metamaterials with a twist, Science 358, no 11 (2017): 1072–1074
[LEH18] K. Lehner, A. Kalteis, Z. Major, Modelling and simulation of lattice structures using various material models for polymeric materials, Acta Polytechnica CTU Proceedings18 (2018), 48-54

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862015