



ÉCOLE  
**CENTRALE** LYON



PSL



CMM



## INTERNSHIP REPORT

APRIL 2022 - OCTOBER 2022

# Soil-Mixing materials modelling

**Student (ECL)**  
Victor LUDVIG

**Mines Paris**  
François WILLOT  
Petr DOKLADAL

**Univ. Eiffel:**  
Alain LE KOUBY  
Myriam DUC

## Table des matières

<b>I Summary</b>	<b>3</b>
<b>II Reconstruction of 3D material based on 2D cross sections</b>	<b>3</b>
II.1 Introduction . . . . .	3
II.2 Work base . . . . .	4
II.3 Enhancement of the work and correction of issues . . . . .	4
II.4 Discussing the simulated material . . . . .	4
II.5 Preprocessing of data fed to autoencoder . . . . .	8
II.6 Modified cross entropy loss . . . . .	9
II.7 Spectral decomposition . . . . .	9
<b>III Analysis of mechanical response of simulated 3D microstructures</b>	<b>9</b>
III.1 Introduction . . . . .	9
III.2 Mechanical formalization . . . . .	10
III.3 Homogenization theory . . . . .	11
III.4 Computation method . . . . .	11
III.5 Mohr-Coulomb model . . . . .	11
III.6 Material dataset . . . . .	12
III.7 Material properties . . . . .	14
III.8 Computation of the mechanical response . . . . .	15
III.9 Analysis of the mechanical response . . . . .	17
III.10 Extreme cases . . . . .	18
III.11 Prediction of plastic behavior based on elastic slopes . . . . .	18
III.12 Comparison with the ground-truth material . . . . .	23
III.13 Estimation of $R_c$ based on volumetric fraction . . . . .	25
<b>IV Supervised Learning for mechanical parameters</b>	<b>27</b>
IV.1 Introduction . . . . .	27
IV.2 Dataset . . . . .	28
IV.3 Suport Vector Regression (SVR) . . . . .	28
IV.4 Multilayer perceptron (MLP) . . . . .	29
IV.5 Results . . . . .	29
<b>V Appendix</b>	<b>30</b>

---

V.0.a	Adapted TL-Embedding network	30
V.0.b	Numerical elasticity matrices of the 3 phases (MPa)	31

## I Summary

In this 6-month research internship, we simulate 3D microstructures of Soil-Mixing materials. These materials consist in the mixing of several types of materials in a full-fledged 3D space, which prevents the use of analytical solutions. Hence, machine learning methods are used to better model the microstructures.

My first task was to improve the results of a past 6-month research internship. The goal was to reconstruct a 3D microstructure based on images of 2D cross-sections. A TL-embedding neural network [4] was used to produce 3D shapes based on 2D cross-sections. However, some issues such as the oval shapes of the generated forms prevented the work from being published. I improved the 3D autoencoder of the neural network for complex 3D voxel forms, implemented with PyTorch. I built PyTorch/NumPy algorithms to compute fast 3D rotations of 3D fields, in order to pre-process / post-process data. Other 3D transformations were also implemented.

Then I implemented supervised learning algorithms (SVR-MLP), to predict mechanical parameters with a regression (see part IV). I used scikit-learn [11] and methods proposed in research-papers [2] to fine-tune the parameters. I had to implement a user-friendly interface so that researchers of Univ. Eiffel University could easily use my algorithms. My last and foremost task was to implement machine learning algorithms to process the mechanical response of artificial 3D microstructures, to better understand the impact of inclusions localizations on mechanical behavior (see part III). The results of artificially generated microstructures were then compared with the mechanical response of the ground-truth material. I showed that the elasticity threshold (highly non linear) can be predicted with good accuracy with linear parameters, and more substantially, that the accuracy is *improved* when more elasticity parameters are added (see part III.11).

## II Reconstruction of 3D material based on 2D cross sections

### II.1 Introduction

This work was the sequel of Louis Le Prince research internship, and I will focus on my contributions. I improved the algorithm to generate an artificial 3-phase microstructure, and I developed an algorithm to pre-process data before they are fed into an autoencoder. This autoencoder is part of a TL-embedding network [4]. The architecture chosen by L. Le Prince is shown section V.0.a.

The microstructures are generated by randomly inserting a rock and mud inclusions in a 3D voxel cubic space. The 3D unfilled space represents the homogeneous field which contains the inclusions.

The inclusions are chosen in the same data set for rocks and mud. The 15.000 forms data set was generated in a past research internship, by augmenting a small public data set of 250 shapes. The initial forms were first clustered according to their spherical harmonic decomposition, and then union, intersection and/or rotations were applied

between forms evenly chosen among the clusters to generate the final data set. This process ensures a diversity of forms in the generated data set.

## ***II.2 Work base***

Louis Le Prince developed an algorithm to generate cylindrical artificial microstructures based on 2D cross-sections of the ground-truth material. The nature of the microstructure is presented in section III.

## ***II.3 Enhancement of the work and correction of issues***

I made several enhancements to Louis Le Prince work, including :

- Improvement of volumetric fractions computations with the computation of covariances and crossvariances on the 3D cylindrical structure.
- Improvement of enlargement of  $50^3$  shapes.
- Changed the code so that it could generate cubic microstructures in order to make mechanical computations with François Willot's code.
- Implemented periodical boundary conditions in order to make mechanical computations.
- Corrected a boundary issue in the position of inclusions inside of the cylinder.

## ***II.4 Discussing the simulated material***

Louis Le Prince did not compare his artificial microstructure with the ground-based material it was supposed to model, and I was tasked to make this comparison. Three main characteristics of the microstructure can be used to evaluate the truth of the artificial material :

1. Correct granulometry, i.e. the artificial microstructure contains inclusions of the right sizes. This parameter was directly governing the generating algorithm and therefore was correct.
2. Correct covariances, i.e. spatial repartition of the inclusions.
3. Correct inclusion shapes.

Considering a voxellised material of volume  $V$ , a material and a phase of characteristic functions  $\chi$  and  $\chi_p$ , the covariance along  $x$  axis, noted  $C_x$ , can be defined as :

$$\begin{aligned} C: \mathbb{N} &\rightarrow [0, 1] \\ n &\mapsto \frac{\int^x \chi_p(x)\chi_p(x+n)dx}{\int^x \chi(x)\chi(x+n)dx} \end{aligned}$$

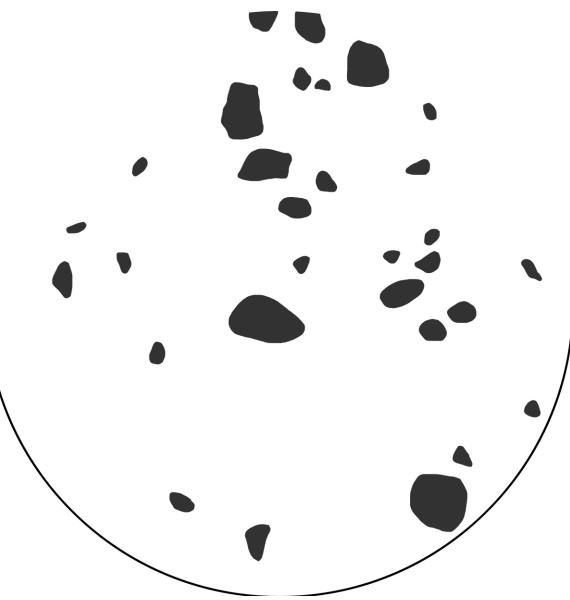
In simple words  $C_x(n)$  is the probability that two points at a distance of  $n$  voxels belong to the same phase.

In particular,  $C_x(0)$  is the volumetric fraction of the studied phase.

I computed covariances for the two types of inclusions inside of the material. Around 50 hand-labelled cross-sections image of each of the 2 inclusion phase of the ground-truth material were sent by Univ. Eiffel (fig 2a). An example is shown below figure 2b.



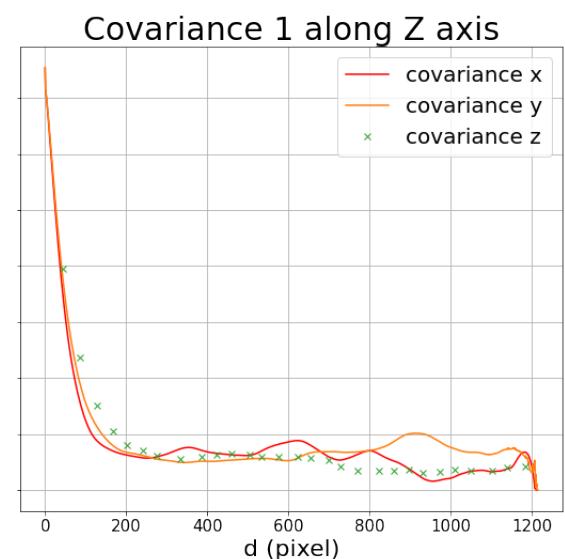
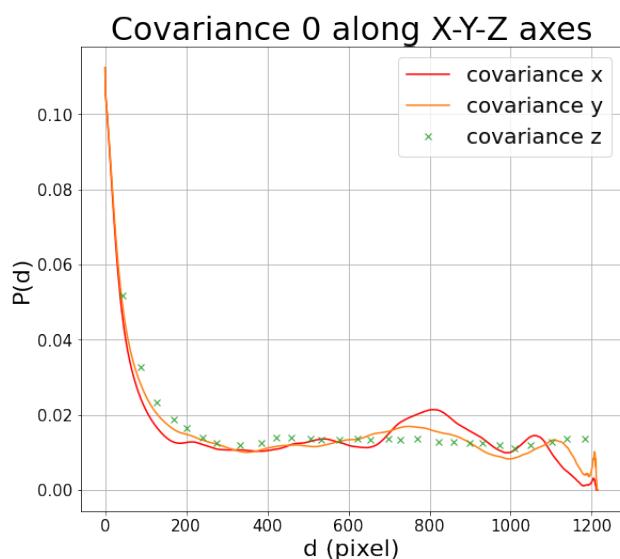
(a) Ground-truth



(b) Labelled cross-section

**FIGURE 2 –** A sample of the ground-truth material was sliced into cross-sections (fig 2a), which were then hand-labelled by Univ. Eiffel (fig (2b))

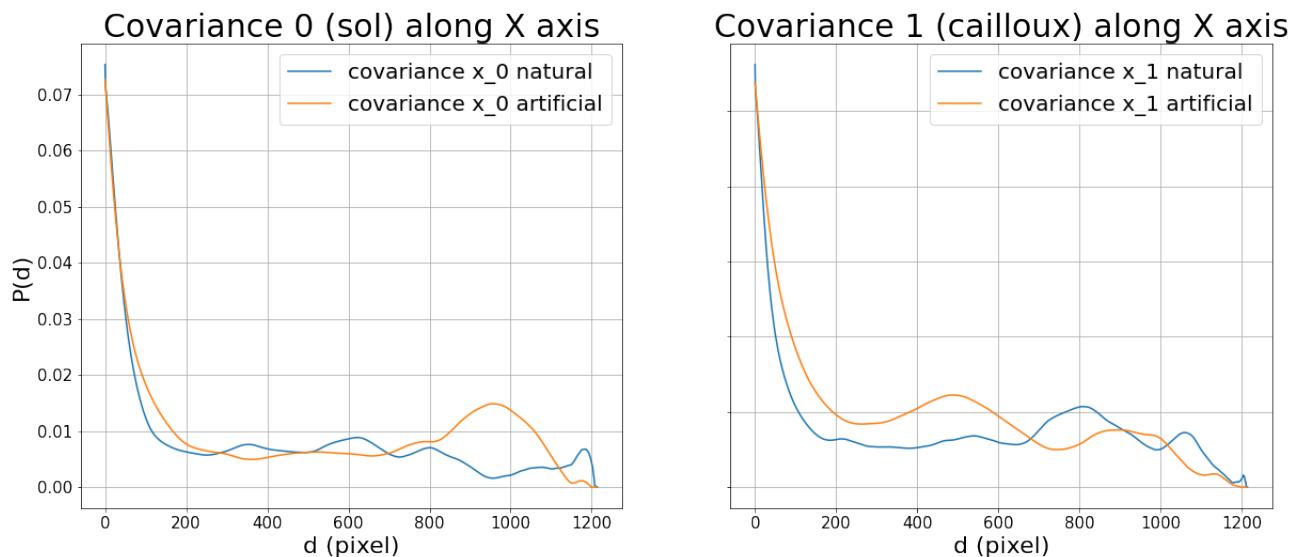
Covariances along X-Y-Z axes for both phases are shown below :



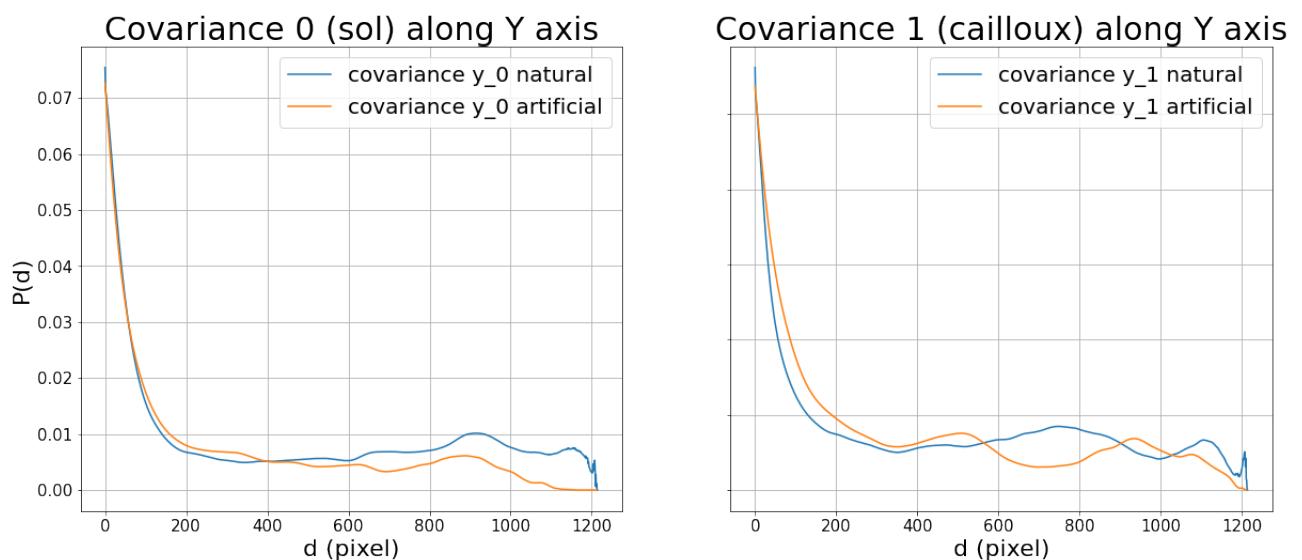
**FIGURE 3 – 0 : Mud, 1 : Rock**

Figure 3 shows that for small distances, the ground-truth material is isotropic. Hence, the method used by Louis Le Prince which relied on isotropy to generate the artificial material is correct. Note that the dotted line for the  $C_z$  stems from the discretization of the material along Z axis, because cross sections were sent. From a technical point of view, the distance between each cross section had to be taken into account for the  $C_z$  computation. Two cross sections had a variable distance between them which were saved in a excel file that Univ. Eiffel provided.

Now we can compare the ground-truth covariances of rock and mud with the covariances of the simulated material.



**FIGURE 4 –** Comparison of ground-truth and simulated covariances along X axis



**FIGURE 5 –** Comparison of ground-truth and simulated covariances along Y axis

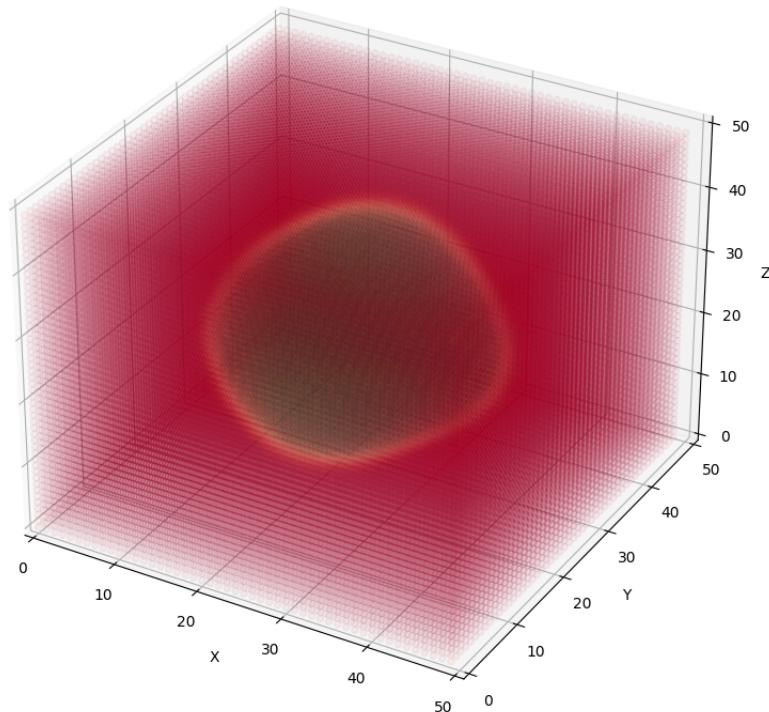
Figure 4 and 5 show that mud inclusions have the same shapes and distributions for

both the ground-truth and simulated material. A small difference can be seen for rock inclusions along both X and Y axes. It seems that rock inclusions of the ground-truth material are a bit smaller than the rocks inside of the simulated material. I did not correct this small issue of Le Prince generating algorithm.

The discrepancy observed for larger distances probably stems from the small number of points at high distance, and was not taken into account.

Overall the computation of covariances showed that the simulated material had a similar distribution of inclusions.

The third point to check is the shape of individual inclusions. A quick review of the form outputted by Le Prince's neural network showed that the overall shape was too oval. Figure 6 shows an example of a 3D probability field outputted by the TL-embedding network that I trained. This 3D inclusion are then obtained by applying a threshold to the probability field. I had to leverage the transparency alpha channel in order to plot the 3D field probability field. We tried to correct the oval shape issue and the corresponding work is discussed in the next section.



**FIGURE 6 –** Probability field outputted by the neural network. Red = low probability, blue = High

## II.5 Preprocessing of data fed to autoencoder

The idea was to preprocess the 3D images fed to the autoencoder of the TL-embedding network. In fact, two identical 3D images can have a very high point wise distance between them if their orientations is not the same.

A simple way to normalize the orientation is to align rotation axes with the orthonormal basis (x,y,z). The rotation axis are defined as the eigen vectors of the inertia matrix I. By definition, with  $p_v$  = probability of the voxel  $v(x_v, y_v, z_v)$  ( $p_v = 1$  for binary images), the inertia matrix reads :

$$\begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{pmatrix} = \begin{pmatrix} \sum_v p_v(y_v^2 + z_v^2) & -\sum_v p_v x_v y_v & -\sum_v p_v (x_v + z_v) \\ -\sum_v p_v (y_v + x_v) & \sum_v p_v (x_v^2 + z_v^2) & -\sum_v p_v y_v z_v \\ -\sum_v p_v z_v x_v & -\sum_v p_v z_v y_v & \sum_v p_v (x_v^2 + y_v^2) \end{pmatrix} \quad (1)$$

The inertia matrix is real and symmetric, hence the existence of real eigen values is assured by Spectral Theorem.

Once the eigen vectors are calculated, they are ordered from biggest to lowest eigen value, and then aligned respectively with (x,y,z).

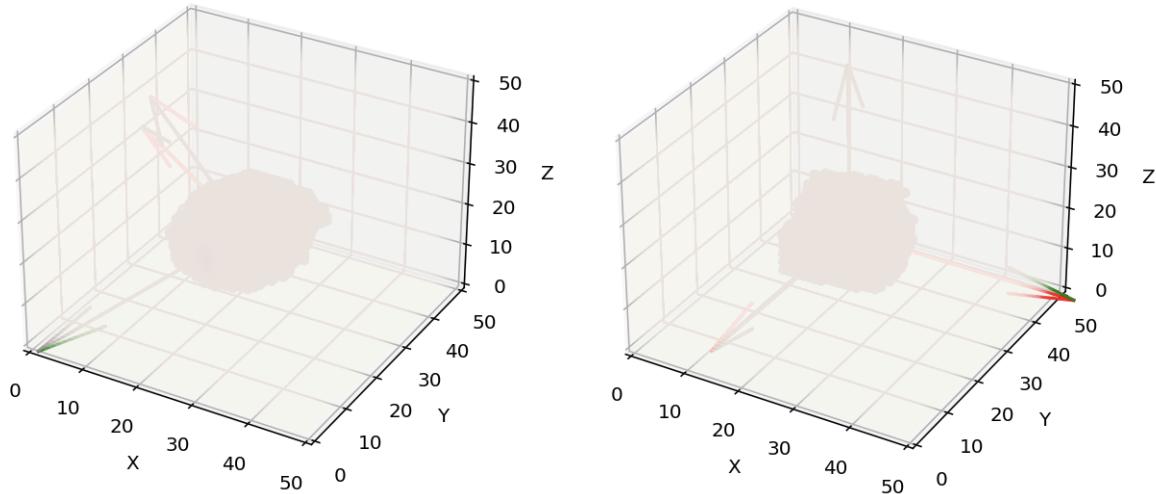
Rotations are performed using rotation matrices. Knowing axis of rotation n and angle a, with  $c = \cos(a)$  and  $s = \sin(a)$ , the corresponding rotation matrix is given by ([12]) :

$$M = \begin{pmatrix} c + (1 - c)n_1^2 & (1 - c)n_1n_2 + sn_3 & (1 - c)n_1n_3 - sn_2 \\ (1 - c)n_1n_2 - sn_3 & c + (1 - c)n_2^2 & (1 - c)n_2n_3 + sn_1 \\ (1 - c)n_1n_3 + sn_2 & (1 - c)n_2n_3 - sn_1 & c + (1 - c)n_3^2 \end{pmatrix} \quad (2)$$

Computations are performed in the pre-processing phase, but we also tried to perform them in the post-processing phase, i.e. after the 3D image has been outputted by the autoencoder. Hence, I implemented the normalization algorithm in both pytorch and numpy.

Other technical details are necessary in order for the computation to be efficient. As an example I leveraged numpy/pytorch vectorization to fasten the computation. The algorithm as well as detailed explanations can be found on my github [7].

Figure 7 is an example of a normalization of a 3D binary image (from my github page). The blurring effect comes from the fact that I leverage the alpha (transparency) channel in order to plot the whole 3D field, as in figure 6.



**FIGURE 7 –** Before (left) and after (right) orientation normalization

## II.6 Modified cross entropy loss

In order to get more detailed shapes output by the autoencoder, a higher number of false-positive voxels can be tolerated. On the contrary, false-negative should be limited. Therefore, I tried to modify the cross-entropy loss, using a weighted cross-entropy, penalizing 100 times more the false negatives than the false positives. However, it did not improve the results.

## II.7 Spectral decomposition

We thought of using spherical harmonics spectral decomposition of 3D shapes to compute the loss, rather than a point-wise loss. I read the original paper [13] and then used a version implemented by a former PhD student and tried to use the spectral decomposition to compute the loss. However, the values of the decomposition can have a dozen orders of magnitude of difference, and therefore I was unable to use it. The theory behind spherical harmonics is also incomplete and scientists working on it are still trying to figure out how the decomposition is related to the actual 3D shape.

# III Analysis of mechanical response of simulated 3D microstructures

## III.1 Introduction

In this section, we simulate 3D samples of deep soil mixing materials. This kind of material is composed of the mixing of in situ soil with a binder (e.g. cement). The

in situ soil contains inclusions such as rocks and mud, which changes its mechanical behavior compared to traditional homogeneous materials. Simulations of deep soil mixing material in a 2-dimensional spaces showed that localization of inclusions and volumetric fractions had a strong influence on the strength and stiffness of the material [14][9][3]. This study aims at confirming these results in a 3-dimensional space. 3D simplified models of the ground-truth material are first generated, and then used to analyse the plastic behavior of the material. All of the code is available on my github [6].

### III.2 Mechanical formalization

The goal of the simulation, like in any classical mechanical test, is to find a relationship between the strain  $\sigma$  and the deformation  $\epsilon$ . Either  $\sigma$  or  $\epsilon$  is set, and the mechanical model leads to the other parameter.

Though in the elastic phase the relationship between the strain  $\sigma$  and the deformation  $\epsilon$  is linear, it becomes non linear in the plastic phase. As a consequence, the response of the material can only be computed incrementally. The plasticity behavior of the material and associated physical laws (Mohr-Coulomb model) are described part III.5

The problem to solve can be formally expressed as follows :

*Knowing at time  $t$  the strain field  $\sigma_{ij}(x,t)$  and the displacement field  $u_i(x,t)$ , their variations are described by :*

- incremental equilibrium equations :

$$\dot{\sigma}_{ij,j} + \dot{f}_i = 0 \quad (3)$$

where  $f$  is the volumetric forces vector (e.g. weight).

- boundary conditions :

$$\dot{\sigma}_{ij}n_j|S_f = \dot{T}^d \quad (4)$$

$$\dot{u}_i|S_u = \dot{u}_i^d \quad (5)$$

- the incremental behavior law :

$$\dot{\epsilon}_{ij} = \mathbb{C}_{ijkl}\sigma_{kl} + \dot{\epsilon}_{ij}^p \quad (6)$$

where  $\mathbb{C}$  is the elasticity tensor and  $\dot{\epsilon}_{ij}^p$  is the temporal derivative of the plastic deformation.

Equation (1) is only verified if the process is quasi-static, i.e. an equilibrium state is found after each incremental step. This condition is verified by the computation algorithm which uses very small loading steps.

The only volumetric force is the weight of the material which is neglected compared to the external load. Hence, equation (1) can be rewritten as :

$$\dot{\sigma}_{ij,j} = 0 \quad (7)$$

Boundary conditions (equation (2), (3)) require a little more explanations and are discussed in the next section.

### III.3 Homogenization theory

Though equation 6 makes it possible to compute a 3D tensor field at each incremental step, it fails to provide a simple description of the mechanical response, such as a scalar. A simple scalar representing the deformation could be the mean of the deformation on the 3D space, which is justified when specific conditions are met, courtesy of **Hill-Mandel theorem** :

Let  $\epsilon_{ij}$  be a kinematically admissible strain and  $\sigma_{ij}$  be a statically admissible stress. Then the three types of boundary conditions :

- $u_i = \bar{\epsilon}_{ij}x_j$
- $\sigma_{iq} = \bar{\sigma}_{iq}n_q$  on  $\partial S$
- $u_i = \bar{\epsilon}_{ij}x_j$

lead to the macro-micro energy equivalence :

$$\frac{1}{V} \int_S \sigma_{ij}\epsilon_{ij}dV = \bar{\sigma}_{ij}\bar{\epsilon}_{ij} \quad (8)$$

In other words, when one of these three conditions are met, the mean of  $\epsilon$  and  $\sigma$  characterize the material mechanical response. Though the material energy is not used in our study, condition (3) of the Hill-Mandel theorem is verified. In fact, François Willot showed that periodical boundary conditions on a finite 3D material lead to a mechanical response closest to that of an infinite material, hence closest to the ground-truth material response. The computation method also relies on peridiodical method, as explained in next section.

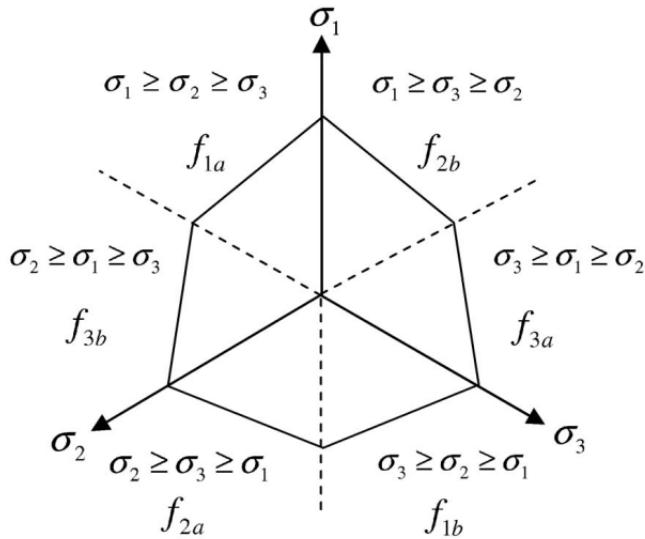
### III.4 Computation method

The 3D Mohr-Coulomb 3D mechanical problem is solved using the FFT method. It was implemented in Fortran by François Willot. This method has better convergence with periodical boundary conditions. For all of these reasons, periodical boundary conditions are used to solve the problem.

Hence, the problem is completely defined by setting a periodical deformation as boundary condition (2).

### III.5 Mohr-Coulomb model

The Mohr-Coulomb model gives a criterion for the material to become plastic. This criterion is based on the values of the stress in the 3 mains directions. It can be represented by the picture shown figure 8.



**FIGURE 8 –** Mohr Coulomb surfaces

Under the Mohr-Coulomb model, a material remains elastic as long as its representation in the 3D stress field is inside the volume represented figure 8. Once the surface is reached, the material becomes plastic, and its representation in the strain field can go in any direction as long as it stays on the Mohr-Coulomb surface. The phenomenon is therefore highly non linear, and supposedly hard to predict. François Willot's algorithm checks at each step whether or not the material is still inside of the Mohr-Coulomb volume, and backtrack if necessary. At first we were not sure that the algorithm would work, and I spent a lot of time on testing.

The Mohr-Coulomb surfaces are defined by some physical parameters, the angle of friction  $\phi$  and the cohesion  $C$ . The physical parameters used for the simulations are displayed table 2. Without going into details, geometric arguments lead to the following equation for the Mohr-Coulomb criterion :

$$f = \frac{1}{2}(\sigma_{max} - \sigma_{min}) + \frac{1}{2}(\sigma_{max} + \sigma_{min})\sin\phi - c\cos\phi \quad (9)$$

François Willot's algorithm checks at each step that this criterion is respected.

### III.6 Material dataset

The problem being set, a dataset of material samples needs to be generated. In the previous internship, a dataset of  $15.000 \cdot 50^3$  images of rocks and mud was generated. In order to compute a mechanical response over the 3D field, the material has to be represented by a point cloud.

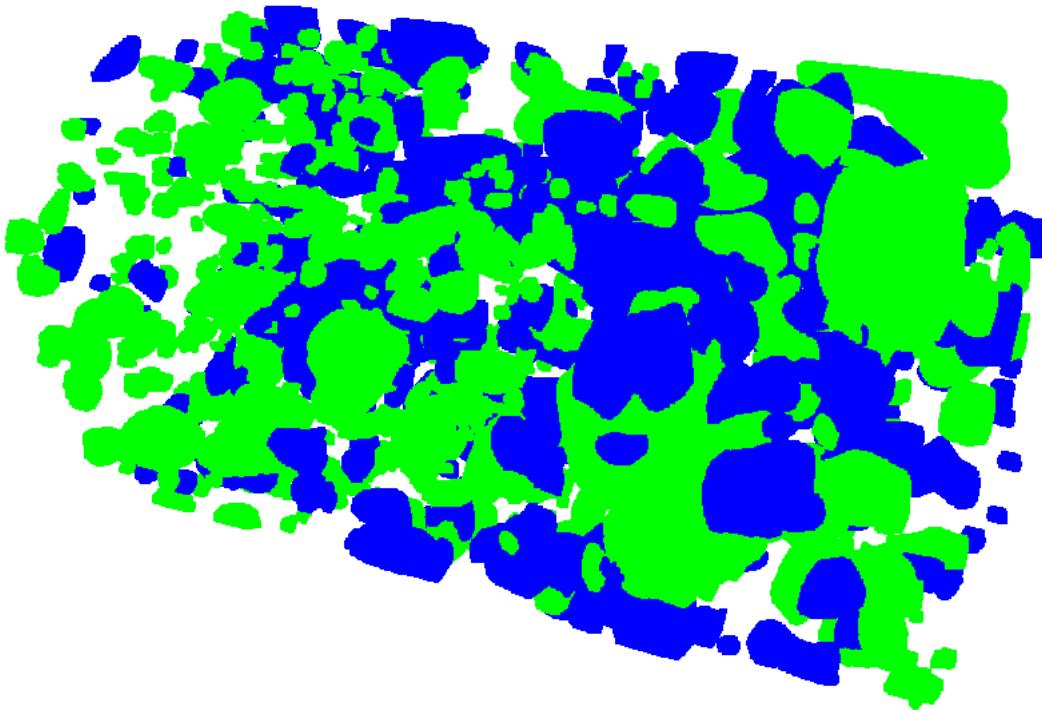
The goal is now to generate 3D images of Soil-Mixing material which respect boundary conditions and share some inclusion properties with the ground-truth material. The most straightforward property of the ground-truth material is the volumetric fraction of rocks and mud, hence the samples will be generated with volumetric fractions close

the ground-truth volumetric fractions.

A sample of the ground-truth material was provided by Univ. Eiffel researchers, which we converted into a numpy array. After some transformations, setting rocks in green and mud in blue, the 3D ground-truth material looks like in figure 9. As stated above, the material is represented by a point cloud. Considering the dimensions of the material (398, 253, 254), more than 25 million points have to be rendered. Hence, the gpu has to be leveraged. Matplotlib does not support gpu usage. As a consequence, I learnt to use the python library vispy to render 3D points cloud. As the library is still in its infancy, not many documentations were available. Hence, I read hundreds of issues and solutions on vispy github page to learn to use it. I also had to struggle a bit to get the nvidia drivers to work.

$f_{rock}(gt)$	$f_{mud}(gt)$	$N_{rock}$	$N_{mud}$
0.0531	0.0803	3	4

**TABLE 1** – Parameters used to generate the dataset

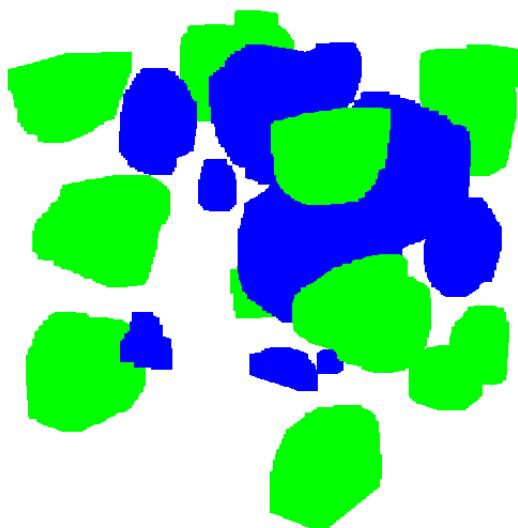


**FIGURE 9** – Sample of the ground-truth material, (398, 253, 254), green=rock, blue=mud

The volumetric fractions of the ground-truth material was provided by Univ. Eiffel. I checked the result and used the ground-truth numpy array to compute the volumetric fractions. As the 3D images of rocks and mud have dimensions of  $50^3$ , I decided to generate  $100^3$  materials. François Willot algorithms work better with cubic microstructures, hence the cubic dimensions. However, as explained in section III.12, the mechanical

response of cylindrical materials can also be computed.

Then we considered two options to generate the dataset. The first consisted in computing the average volumetric fractions of rocks and mud of the  $50^3$  datasets, then in using this volumetric fractions to compute the right amounts of rocks and muds to put in the material, and then use this number as the parameter of a Poisson law to add some variability to the model. However, it was found that this approach led to a too high variability in the volumetric fractions of the generated materials. Hence I used a second method : the right number of rocks and mud was computed as in method one, and then directly used to generate the materials. The parameters used and found in this method are summarized in table 1. An example of a generated for the dataset is shown figure



**FIGURE 10 –** 3D periodic artificial points could material,  $100^3$ , with 3 rock (green) and 4 mud (blue) inclusions. 3D image generated with visipy.

### III.7 Material properties

Once the material dataset has been generated, the material parameters have to be computed to make the simulations. Elasticity matrices and cohesion coefficients of the matrix material, rocks and mud have to be determined. The elasticity matrix is given by the Hooke's law for isotropic materials :

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{yz} \\ 2\epsilon_{xz} \\ 2\epsilon_{xy} \end{pmatrix} \quad (10)$$

with  $\nu$  the Poisson coefficient and  $E$  the Young modulus.

Under the Mohr-Coulomb model, the Cohesion  $C$  and the angle of friction  $\phi$  also have

to be determined. These values were provided by Univ. Eiffel, and are shown in table 2.

Phase	E (GPa)	Cohesion C (MPa)	Friction angle $\phi$ (°)	Poisson coefficient $\nu$
Matrix	15.1	10	34	0.25
Mud	0.2	0.02	34	0.4
Rocks	75	30	40	0.25

**TABLE 2 –** Material properties

$E$ ,  $\phi$ ,  $\nu$  of the 3 phaes were determined using values from the scientific literature, as well as  $C$  for mud and rocks.

$C_{mat}$  was determined by Univ. Eiffel using the linear relation between the plasticity threshold  $R_c$  and  $C$ . In fact, experimental results [5] show that the relationship between  $R_c$  and  $C$  can be written :

$$R_c = aC \quad (11)$$

For our material, the literature suggests :

$$a = 0.5 \quad (12)$$

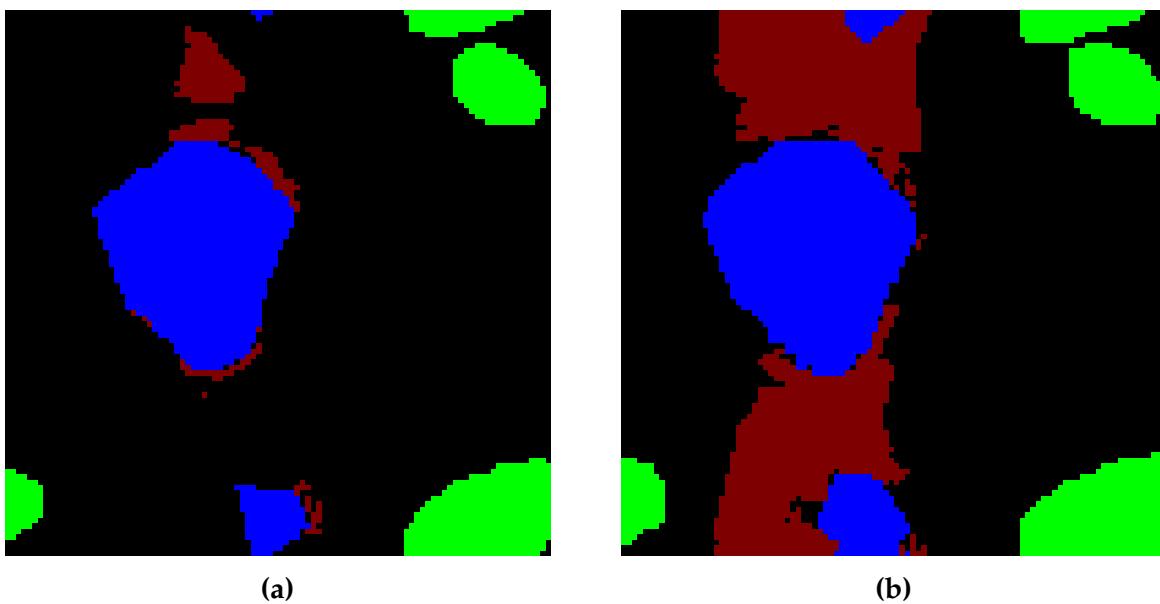
As  $C$  is known for materials with volumetric fractions of mud and rocks between 10% and 20%, the  $a$  factor can be computed by taking the predicted value for a homogeneous material. The  $R_c$  threshold for the matrix was computed by Univ. Eiffel by doing a linear regression on the volumetric fraction of rocks and mud. This method is discussed in part '**Estimation of  $R_c$  based on volumetric fraction**'.

### **III.8 Computation of the mechanical response**

Mechanical properties of materials can be computed either by compressing or pulling the material. Although both configurations are equally hard to simulate, compression is much easier to implement in a lab than traction. So as to compare the results with the ground-truth results from Univ. Eiffel lab, only compression will be simulated.

According to mechanical theory, the compression behavior of a material can be fully determined by its response in the 6 directions  $xx$ ,  $yy$ ,  $zz$ ,  $xy$ ,  $xz$ ,  $yz$ . In this paper, only the mechanical response in the  $xx$  axis is studied, although the study can be very easily extended to the other axes. The material being on average isotropic,  $xx$ ,  $yy$  and  $zz$  are equivalent.

As explained in part III.3.b, the deformation  $\epsilon$  is gradually incremented with a fixed step called *loading\_step* and the strain  $\sigma$  is determined for each value of  $\epsilon$ . At each step, a 3D tensor field is computed for  $\epsilon$  and  $\sigma$ , i.e. 2 3x3 symmetric tensors in each voxel. This requires heavy computations and makes it impossible to model too large micro-structures, or to take too large number of *loading\_step*. After a series of thorough testing, it was found that a *loading\_step* of  $3.10^{-5}$  and a number of steps of 50 lead to good convergence and was enough to observe both elastic and plastic behavior for  $100^3$  volumes.

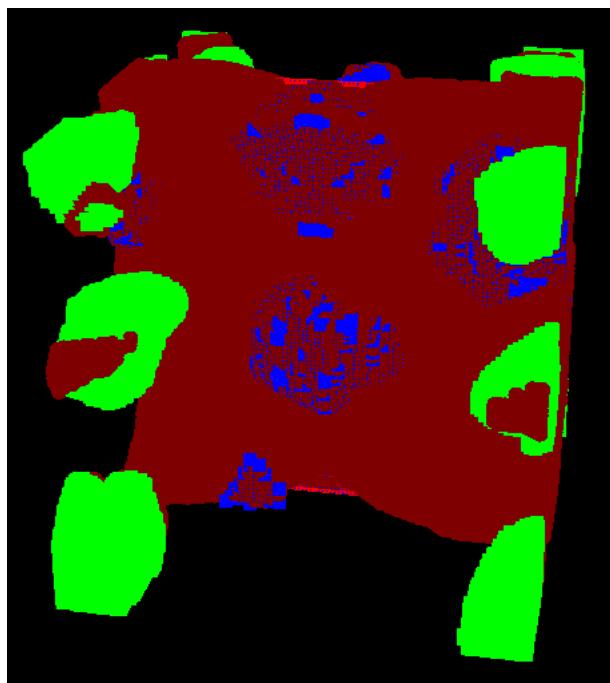


**FIGURE 11** – Cross sections of the 2D material showing in red the deformation field  $\epsilon_{xx}$ .

Computation time is around 1 day with 16 cpu cores for 1000 of  $100^3$  volumes. In this study, 3.000 microstructures are modelled.

The 3D tensor fields computed at each step have to be transformed in order to interpret the results at the end of the experiment. As explained in the homogenization theory part, the state of the material can be computed by taking the means of  $\epsilon$  and  $\sigma$  on the 3D volume. Hence the means of deformation tensor  $\epsilon_{xx}$  and strain tensor  $\sigma_{xx}$  are computed at each step, and then plotted against each other.

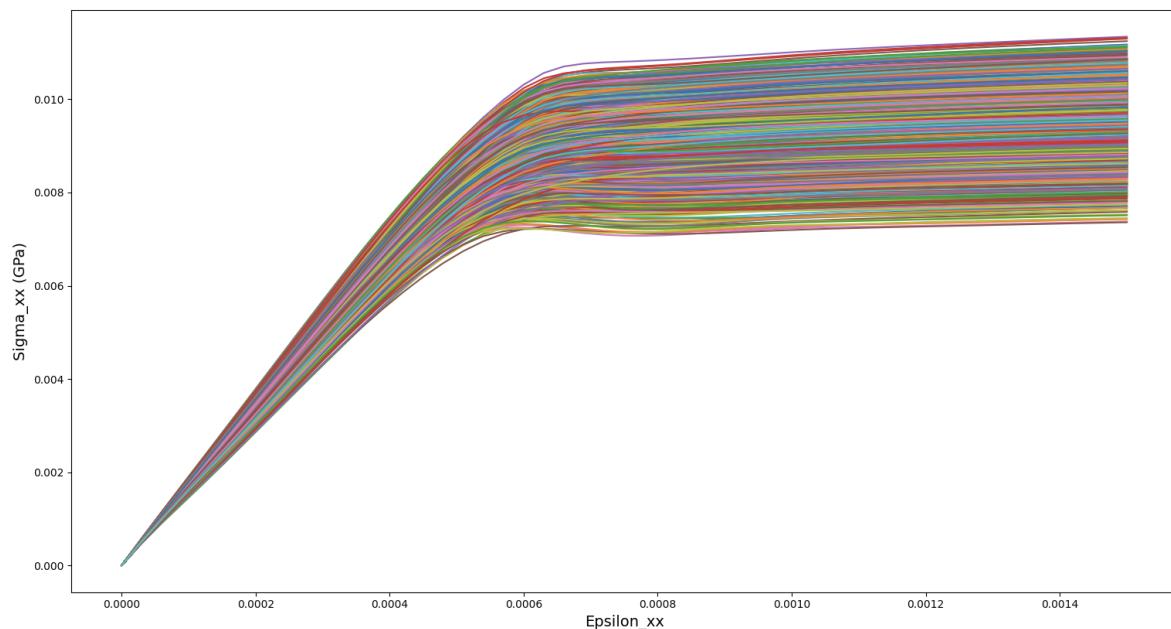
A 2D and 3D visualizations of the deformation field across the 3D volume were implemented. An example of 2D visualisation is given figure 11. See figure 12 for an image of 3D visualization. These figures show that the deformation field is most important around the mud inclusions, and starts around mud. This result was predictable because mud inclusions have the lowest Young modulus.



**FIGURE 12 –** 3D image of the deformation field  $\epsilon_{xx}$ .

### III.9 Analysis of the mechanical response

With a *loading\_step* of  $3.10^{-5}$  and a number of step of 50,  $\epsilon$  ranges from 0 to  $15.10^{-4}$ . Loading curves of 3000 randomly generated materials are plotted below.



**FIGURE 13 –** 3000 loading curves along xx axis

The first linear part of the curve matches the elastic phase. Then the curve becomes

non linear which matches the plastic phase. In a one-phase material, the curve should saturate after the plastic threshold. However, the existence of inclusions inside of the material leads to a linearly increasing slope (in absolute value) after the plastic phase. Though a third constant-sigma phase was reached for microstructures with smaller volumetric fraction of inclusions, increasing by a factor of 100 the loading step was not enough to reach the constant phase for the studies microstructure. Hence the choice of the parameters *loading\_step* and the number of steps for the experiment, which are enough to clearly distinguish the elastic and plastic phases.

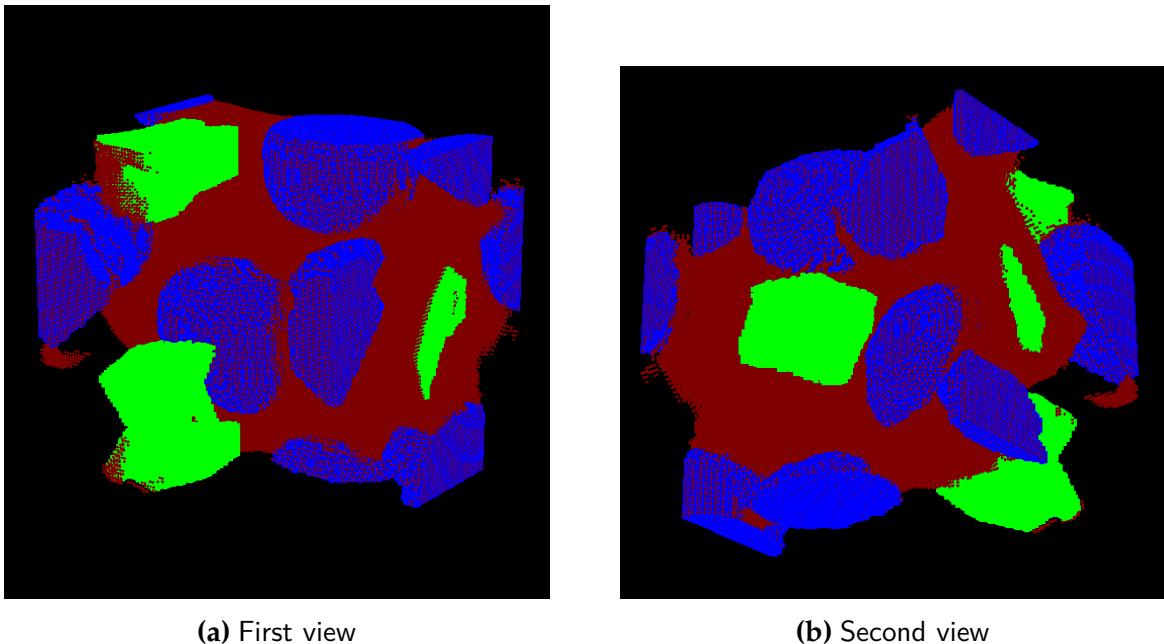
Figure 13 shows that, for a fixed value of epsilon, the strain inside of the material is different for 2 microstructures in the elastic phase as well as in the plastic phase. This example shows that the value  $\sigma(\epsilon = 0.0015) = \sigma_1$  can vary up to 50% from one microstructure to other. This difference stems from different values of volumetric fraction and different relative positions of inclusions.

### III.10 Extreme cases

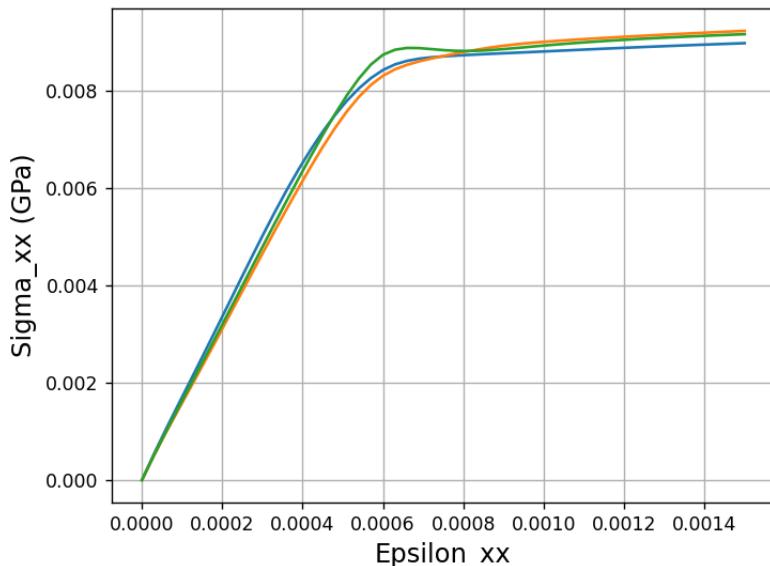
The 3.000 microstructures can be sorted according to their plasticity threshold to highlight the most and least resilient structures. Figure 14 shows 2 views of the least resilient microstructures. We can see a cluster of mud (blue material) at the center of the cube, around which the deformation localization surface is formed. On the contrary, figure 15 shows the most resilient microstructure, which contains a cluster of rocks (green). This example shows that a cluster of rock or mud can greatly influence the resilience of the material at a small scale. Though the dataset was generated using the same number of rock and mud inclusions for each microstructure, these inclusions have different volumetric fractions. Hence, the volumetric fractions of rock and mud is also different for the 2 microstructures ; the least resilient microstructure has a higher volumetric fraction of mud whereas the most resilient microstructure has a higher volumetric fraction of mud. This result was easily predictable, however it proves that the simulation yields realistic results.

### III.11 Prediction of plastic behavior based on elastic slopes

Figure 13 seems to show a correlation between the elastic slope and the plastic threshold  $\sigma_1$ . However some curves are overlapping one-another 19, which indicates a non-trivial relationship. Hence, machine learning models are used in this section to investigate this relationship.



**FIGURE 14 – Least resistant microstructure**

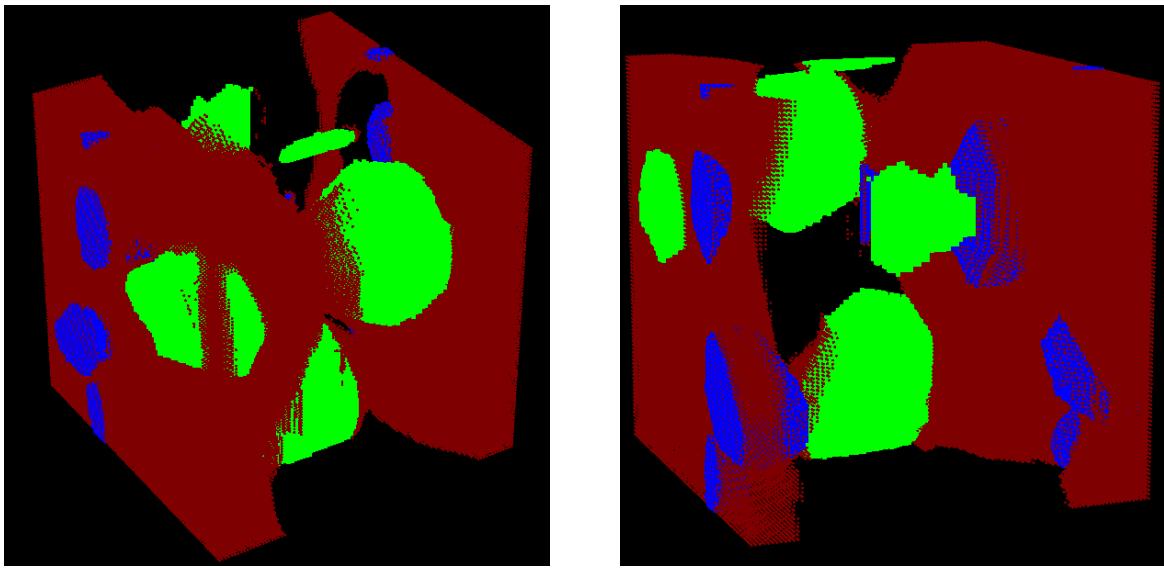


**FIGURE 16 – 3 overlapping curves**

A first simple linear regression is computed using only the slope of the elastic phase as feature. Then the volumetric fractions of rocks and mud  $f_2$  and  $f_3$  are added ( $f_1$  begin the matrix volumetric fraction, i.e.  $f_1 = 1 - f_2 - f_3$ ). To better take into account the 3-dimensional behavior of the material, the elasticity matrix  $C$  is calculated and its unique 21 coefficients are taken as features for a third linear regression. Taking into account homogenization theory, the Hook law can be written :

$$\langle \sigma_{ij} \rangle = C_{ijkl} \langle \epsilon_{kl} \rangle \quad (13)$$

In the elastic phase,  $C$  is a  $6 \times 6$  symmetric matrix and equation (6) reads, using Voigt



(a) First view

(b) Second view

**FIGURE 15 –** Most resilient microstructure

notation :

$$\begin{pmatrix} \langle \sigma_{xx} \rangle \\ \langle \sigma_{yy} \rangle \\ \langle \sigma_{zz} \rangle \\ \langle \sigma_{yz} \rangle \\ \langle \sigma_{xz} \rangle \\ \langle \sigma_{xy} \rangle \end{pmatrix} = \begin{pmatrix} C_{11} & \dots & C_{16} \\ \vdots & \ddots & \vdots \\ C_{16} & \dots & C_{66} \end{pmatrix} \begin{pmatrix} \langle \epsilon_{xx} \rangle \\ \langle \epsilon_{yy} \rangle \\ \langle \epsilon_{zz} \rangle \\ \langle 2\epsilon_{yz} \rangle \\ \langle 2\epsilon_{xz} \rangle \\ \langle 2\epsilon_{xy} \rangle \end{pmatrix} \quad (14)$$

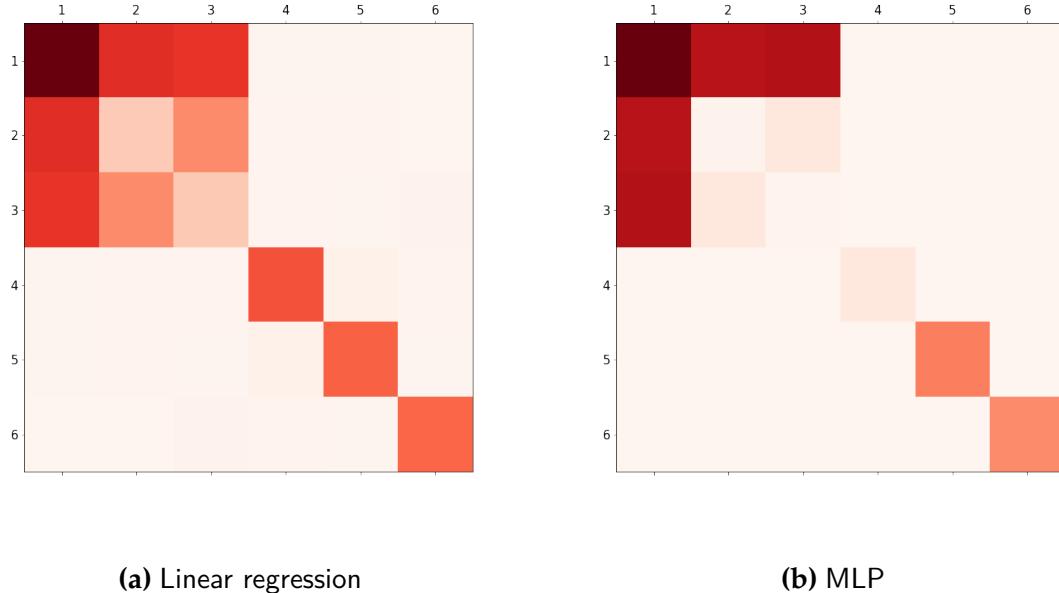
From equation (7), one can see that imposing  $\langle \epsilon \rangle$  in one of the 6 directions and computing  $\langle \sigma \rangle$  leads to one column of the matrix. Hence, 6 computations are required to determine elasticity matrix C.

Linear regressions are computed by taking 80% of the 3.000 samples as training set and 20% as testing set. The score (coefficient of determination) is computed both on the training and validation set. Data is normalized with mean 0 variance 1 before being fed to the regression algorithm. Results are shown in table 3 :

Features	$C_{11}$	$f_2, f_3$	$f_2, f_3, C_{11}$	$f_2, f_3, C$
Number of features	1	2	3	23
Training R <sup>2</sup>	0.464	0.482	0.663	0.7760
Validation R <sup>2</sup>	0.479	0.475	0.651	0.7763

**TABLE 3 –** Linear regression results

Table 3 shows that adding volumetric fractions  $f_2$  and  $f_3$  and more coefficients of the elasticity matrix leads to a better coefficient of correlation. Hence, next regression models take the 23 scalars  $f_2, f_3$  and  $C_{ij}$  as features. To better understand which coefficients of C influence the value of plasticity threshold  $\sigma_1$ , C is plotted in figure



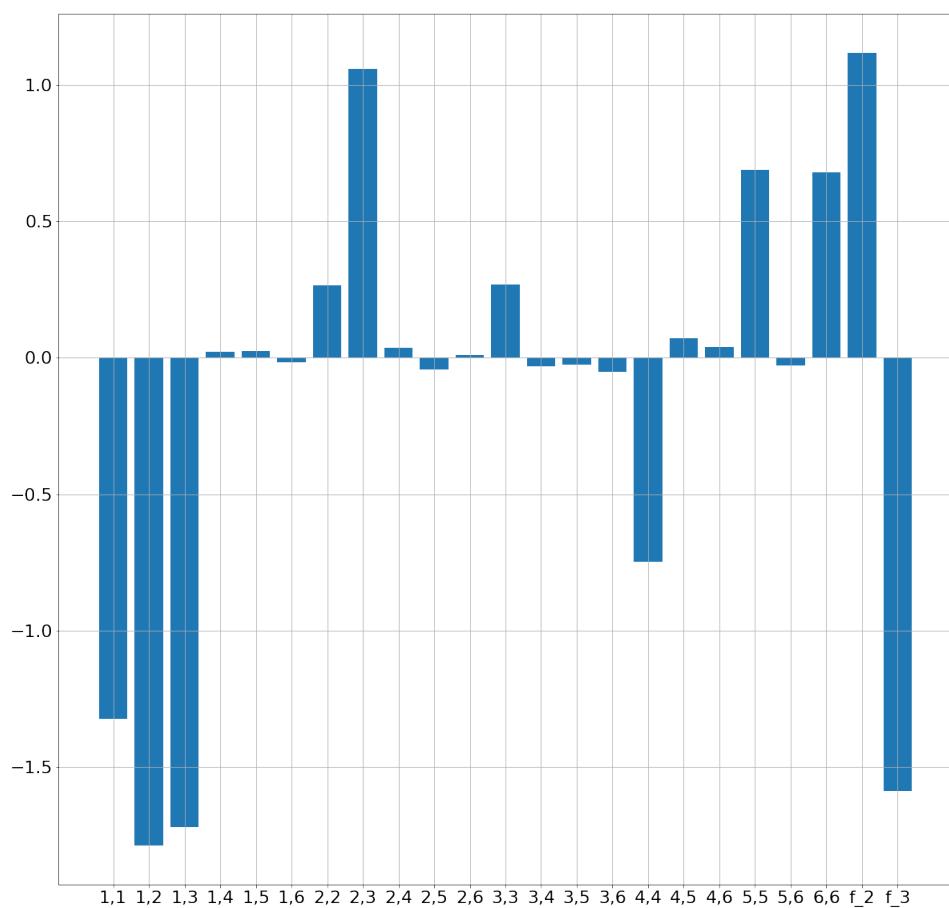
**FIGURE 17 –** Importance of the coefficients of the elasticity matrix on the prediction of the (strongly non linear) elasticity threshold

17a with varying colors according to the absolute value of the related linear regression coefficient. The value of the linear regression coefficients for the normalized matrix coefficients and the normalized volumetric fractions is shown figure 18. As expected,  $C_{11}$  is the most important coefficient, the mud represented by  $f_3$  decreases the plasticity threshold, whereas the rocks represented by  $f_2$  increase the plasticity threshold. This relative influence (positive or negative) can only be computed for the linear regression. For more complex models, only the permutation importance can be computed, which does not tell if the features increases or decreases the target.

Though the linear regression yields very good results, it remains the simplest model. Hence, a multi-layer perceptron (MLP) is trained to encompass the more of the phenomenon's complexity. A grid-search with 5-fold cross validation is first conducted on 80% of the data to find the best number of hidden layers, perceptrons per layer, the activation function and the solver. Data is then trained on the training set and evaluated on the 20% of data left. In order for the model not to overfit and remain relatively simple, the number of hidden layers ranges from 2 to 3 and the number of perceptrons per hidden layer ranges from 1 to 8. Results are shown in table 4. I observed varying hyperparameters through different grid-searches. However, the score was always between 0.76 and 0.80.

Phase	Activation function	Number of hidden layers	Perceptron per hidden layer	$R^2$
Grid-Search	ReLU ; logistic ; tanh ; identity	{2,3}	{8}	0.777
Validation	logistic	2	(3,6)	0.778

**TABLE 4 –** MLP hyperparameters and scores during the grid-search and the validation phases



**FIGURE 18 –** Coefficient of the linear regression on the normalized data

Though the importance of features is not as easy to find as with a linear regression, a permutation importance can be computed, which yields the colored elasticity matrix shown in figure 17b.

The MLP model seems to better discriminate between matrix coefficients. The first coefficient of the matrix (related to axe x) is still the most important, and the 2 most important coefficients,  $M_{12}$  and  $M_{13}$  are the same for the 2 models. One can check that the y and z axes are equivalent, which is expected for an isotropic material compressed along the x axis.

Lastly, all scikit-learn's [11] models are tested with default parameters using the lazy regressor's module [10]. Results show that the linear regression is the best model without parameter tuning, which corroborates the very good results of this simple method compared with the MLP .

		Adjusted	R-Squared	R-Squared	RMSE	Time Taken
1	Model					
2	LinearRegression	0.76	0.77	0.47	0.08	
3	TransformedTargetRegressor	0.76	0.77	0.47	0.01	
5	LassoLarsIC	0.76	0.77	0.47	0.10	
6	RidgeCV	0.76	0.77	0.47	0.04	
7	LassoLarsCV	0.76	0.77	0.47	0.08	
8	BayesianRidge	0.76	0.77	0.47	0.04	
9	Ridge	0.75	0.76	0.47	0.01	
10	KernelRidge	0.75	0.76	0.47	0.20	
11	LassoCV	0.75	0.76	0.47	0.29	
12	LarsCV	0.75	0.76	0.47	0.05	
13	ElasticNetCV	0.75	0.76	0.48	0.27	
14	HuberRegressor	0.75	0.76	0.48	0.09	
15	LinearSVR	0.75	0.76	0.48	0.29	
16	OrthogonalMatchingPursuitCV	0.75	0.76	0.48	0.02	
17	SGDRegressor	0.74	0.75	0.49	0.02	
18	MLPRegressor	0.72	0.73	0.50	2.16	
19	NuSVR	0.72	0.73	0.51	0.29	
20	SVR	0.72	0.73	0.51	0.29	

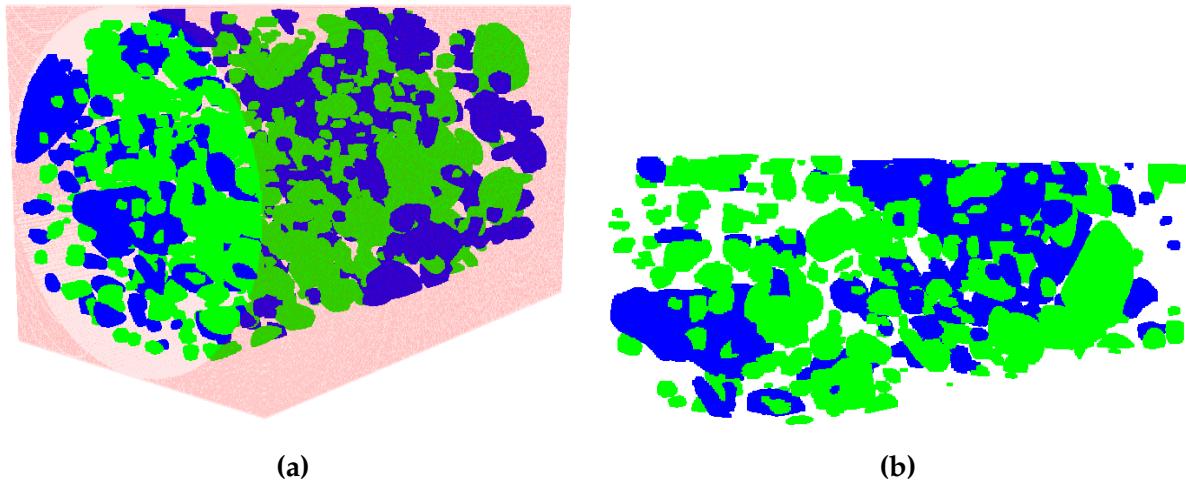
**FIGURE 19 –** 20 best models with default parameters according to LazyRegressor

### III.12 Comparison with the ground-truth material

After this study on a dataset of small periodic materials, we can compare the numerical computations with the results from the lab experimentations on the ground-truth material. The sample of the ground-truth material taken to make the computations is shown figure 9. Though it represents a cylindrical structure, it is represented by a 3D numpy array. Two types of boundary conditions are tested in this study. The first is the whole cylinder with void added outside of the cylinder. The second configuration is a 3D parallelopiped taken inside of the cylinder, with free boundary conditions.

The loading curve is simulated for all of the 3 configurations. From a technical point of view, I had to make sure that the average of sigma and epsilon is taken inside of the cylinder for the cylindrical material with void added on the outside.

Results are shown in figure 25. Though the order of magnitude is correct for the plasticity

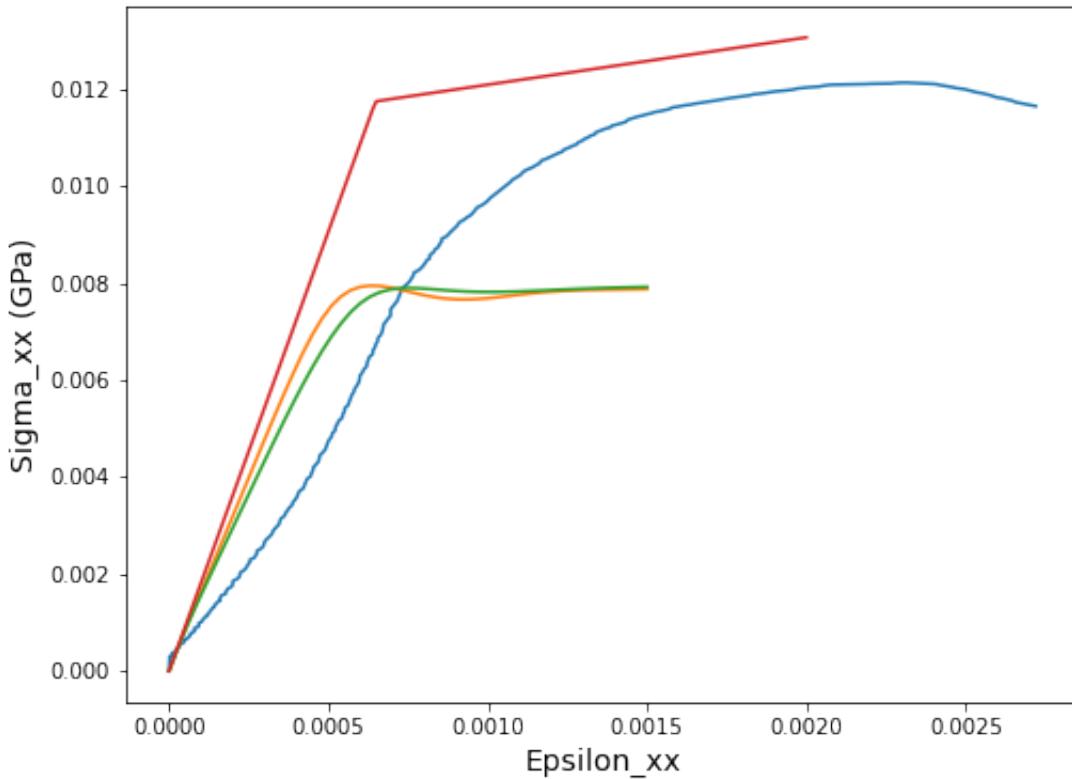


**FIGURE 20 –** Ground-truth material.

(a) : void is added outside of the cylinder.

(b) : a paralleloid is taken inside of the cylinder.

threshold, a small discrepancy exists between the simulated curves and the ground-truth lab experiment. It probably stems from the choice of material parameters, which is discussed in the next section.



**FIGURE 21 – Loading curves**

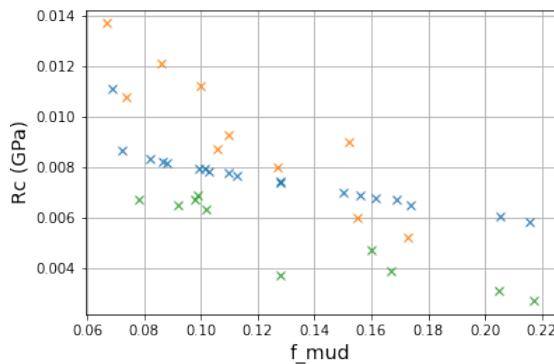
Red : cubic material with only the matrix  
 Blue : lab data of ground-truth material  
 Orange : parallelloid inside of cylinder  
 Green : void added outside of cylinder.

### III.13 Estimation of $R_c$ based on volumetric fraction

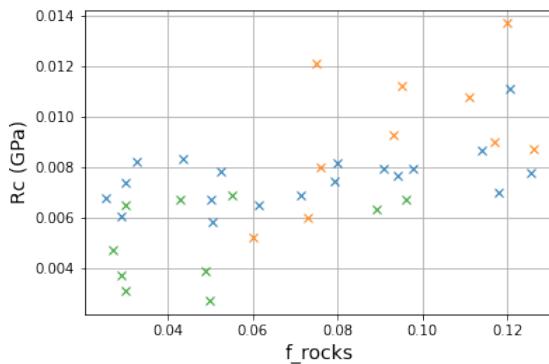
An explained in part , the cohesion factor C for the binding material is computed using a linear relationship which links it to the plasticity threshold  $R_c$ . Hence,  $R_c$  has to be computed for the binding material, although only samples of heterogeneous material are tested in the lab. Hence, a regression based on the volumetric fractions of inclusions needs to be computed to find the  $R_c$  of the homogeneous material.

Figure 22a shows that an affine relationship seems to exist between  $R_c$  and  $f_{mud}$ . This doesn't seem to be the case for the volumetric fraction of  $f_{rocks}$  as shown in figure 22.

This relationship is actually not linear. As shown by François Willot, there exists a power law between  $R_c$  and  $f$ , as the pathological point on figure 22a seems to show. PhD student Juba Amroui from Univ. Eiffel estimated the  $R_c$  coefficient of the matrix material by computing an exponential regression of  $R_c$  and the volumetric fraction of



(a)  $R_c$  against  $f_{mud}$



(b)  $R_c$  against  $f_{rocks}$

**FIGURE 22 –  $R_c$  threshold against volumetric fractions**

Orange and green : experimental results from two materials

Blue : simulation results

mud  $f_{mud}$ . However, it can be difficult based on experimental data to fit the exact power law, which is a factor of error (1).

Another factor of error is the  $a$  parameters of equation 12 (2).

To correct issue (1), a first correction factor between the linear regression and the correct regression on simulated data is computed. The regression is computed on the simulated dataset, and the correct  $R_c$  is computed with a cubic homogeneous matrix material containing 0 inclusion.

Issue (2) is corrected by computing the relation between the  $R_c$  value computed with the regression on the experimental data and the  $R_c$  computed on the homogeneous simulated material.

All in all, issues (1) and (2) are corrected by multiplying the cohesion by the factor between the plasticity thresholds computed with both regressions. The new cohesion factor is  $C = 17.94 \text{ MPa}$ , up 74.94% from the last  $C = 10.00 \text{ MPa}$ . Results are summarized in figure .

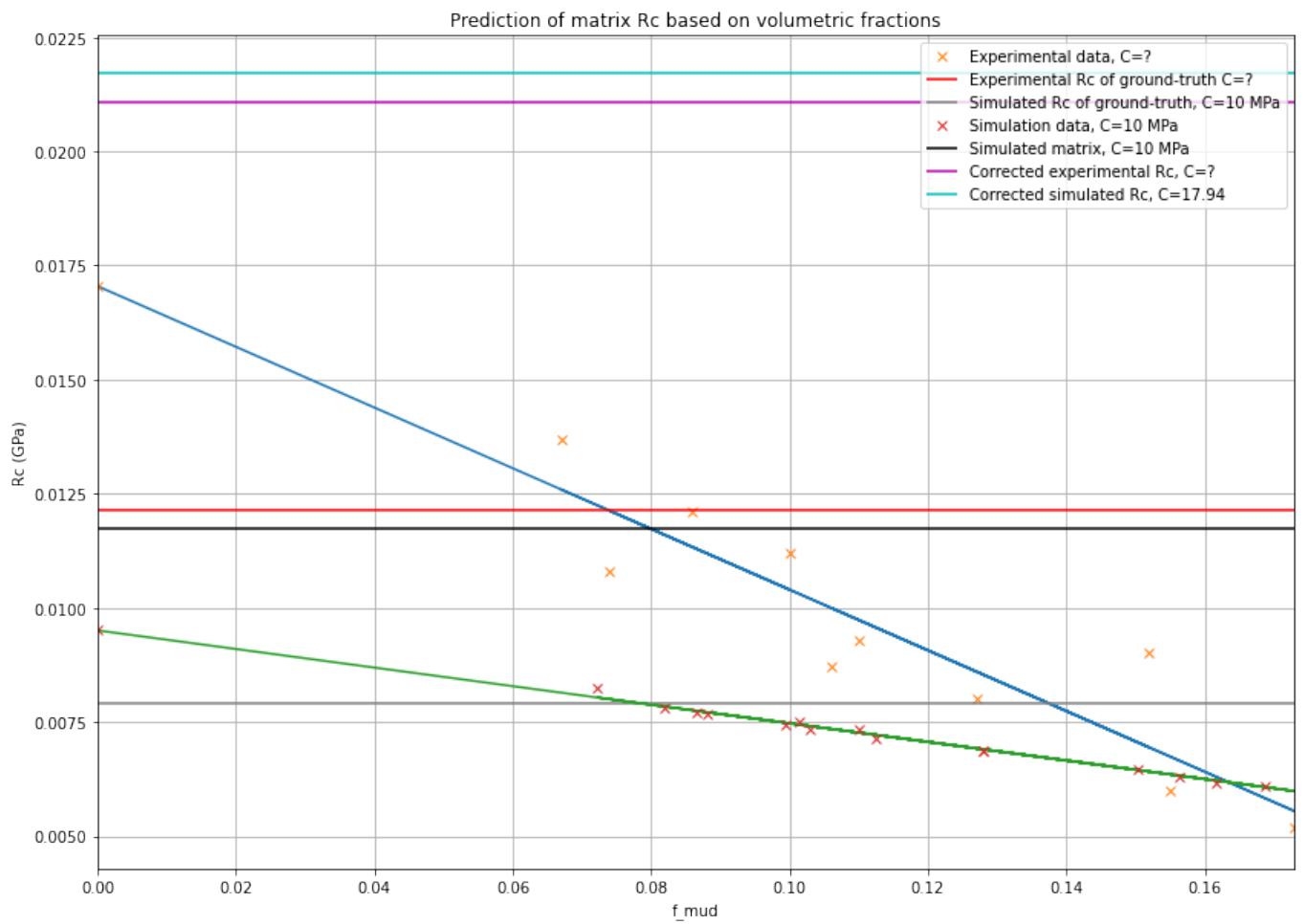


FIGURE 23

One can see that the corrected simulated  $R_c$ , computed numerically on a homogeneous material, is close to the experimental  $R_c$ . There is still a small discrepancy, because the correction factor (1) is not perfect and the other mechanical parameters were taken from the literature and not empirically tested. However, the figure shows that better results can be obtained by adjusting the mechanical parameters.

This work is independent of the 2 other parts. However, the same supervised learning tools used in part III.13 are used.

## IV Supervised Learning for mechanical parameters

### IV.1 Introduction

Inclusions inside of Soil-Mixing materials lead to complex physical behavior which are hard to predict by classical mechanical theories. Hence, classical machine learning methods have been used to predict mechanical parameters of Soil-Mixing materials [1]. The goal of this section is to replicate these methods on new data provided by Univ.

Eiffel, and compare the results. Two regression methods will be tested : Support Vector Regression (SVR) and Multi-Layer Perceptron (MLP). All of the code is available of my github : [8].

## IV.2 Dataset

The dataset is very small and contains only 189 samples. The scarcity of data stems from the difficult lab-experiment conducted. As an example, one parameter is obtained by heating the material for 30 days, which takes time and is prone to many experimental errors.

## IV.3 Suport Vector Regression (SVR)

SVR regression consists in mapping the input space onto a higher dimensional space using a non-linear mapping, and then constructing a linear regression in this new feature space. This mapping is computed by the kernel trick. We use the gaussian kernel (rbf), the most commonly used non linear kernel, defined for  $x$  and  $x'$  in the initial feature space by :

$$k(x, x') = \exp(-\gamma ||x - x'||^2) \quad (15)$$

The gamma parameter, defined  $> 0$ , is chosen with the methodology proposed by Cherkassky [2], as well as C and epsilon.

For an explanation of the problem solved to compute the SVR model, see scikit-learn documentation.

First epsilon is determined. With statistical arguments, we can propose when the number of samples is small (the case here, 189 samples) :

$$\epsilon = \frac{\sigma}{\sqrt{n}} \quad (16)$$

The input noise level is by definition the standard deviation of the output  $y$ , i.e.

$$\sigma_i = (y_i - \hat{y}_i)^2 \quad (17)$$

So we have to estimate the output of the SVR model without having the SVR model. A common approach is to fit the data with a low bias model and apply the following formula to estimate the noise :

$$\hat{\sigma} = \frac{n}{n-d} \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i \quad (18)$$

with  $d$  the degrees of freedom of the estimator. The estimator chosen here is the k-nearest-neighbors method, with  $k=3$  [2]. The model complexity is estimated by

$$d = \frac{n}{n^{\frac{1}{5}} k} \quad (19)$$

The C constant determines the tradeoff between model complexity (decreasing C implies a reduced model complexity) and correct prediction (high C). Cherkassky proposes an improved version of the formula proposed by Mattera and Haykin (1999) :

$$C = \max(|\hat{y} + 3\sigma_y|, |\hat{y} - 3\sigma_y|) \quad (20)$$

with sigma the standard deviation of training target.

Finally, gamma is determined with a 5-fold cross-validation grid-search, with C and epsilon values previously determined. This is the method used by Tinoco in his 2009 thesis.

#### **IV.4 Multilayer perceptron (MLP)**

The MLP model consists in several fully connected perceptron layers. Three parameters have to be chosen :

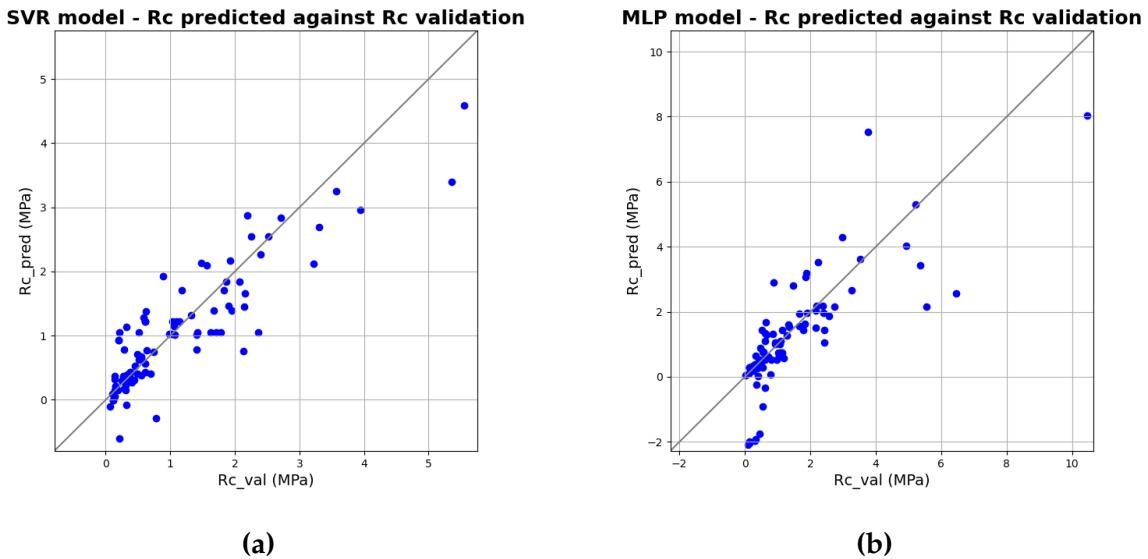
- the number of layers
- the number of perceptron for each layer (may be different from layers to layers)
- the activation function (the same for each perceptron)

These parameters are found doing a grid-search with 5-fold cross-validation on the training set.

#### **IV.5 Results**

Finding an accurate and representative model is particularly hard because of the scarcity of data. In particular, the variability of results is highly dependant on the random subset chosen to train the model. The deterministic choosing of 2 out of 3 of the parameters in the case of the SVR model makes it possible to reduce this variability, which is however still high.

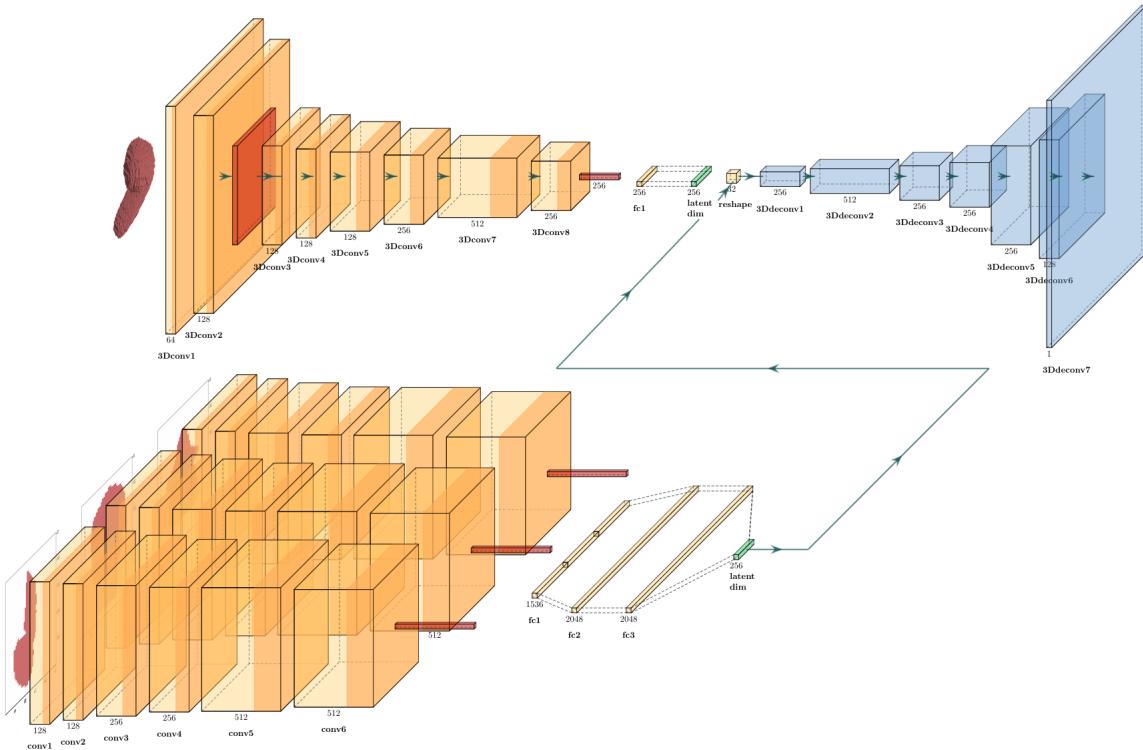
As the parameters chosen vary according to the random subset training chosen, it does not make much sense to present an example which could not be reproduced.



**FIGURE 24 –** Prediction results of  $R_c$

## V Appendix

### V.0.a Adapted TL-Embedding network



**FIGURE 25 –** Adapted version of the TL-embedding network (2016) implemented by L. Le Prince during his research internship.

### V.0.b Numerical elasticity matrices of the 3 phases (MPa)

$$E_{mat} = \begin{pmatrix} 18.12 & 6.04 & 6.04 & 0 & 0 & 0 \\ 6.04 & 18.12 & 6.04 & 0 & 0 & 0 \\ 6.04 & 6.04 & 18.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.04 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6.04 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6.04 \end{pmatrix} \quad (21)$$

$$E_{rock} = \begin{pmatrix} 90.0 & 30.0 & 30.0 & 0 & 0 & 0 \\ 30.0 & 90.0 & 30.0 & 0 & 0 & 0 \\ 30.0 & 30.0 & 90.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 30.0 \end{pmatrix} \quad (22)$$

$$E_{mud} = \begin{pmatrix} 0.4286 & 0.2857 & 0.2857 & 0 & 0 & 0 \\ 0.2857 & 0.4286 & 0.2857 & 0 & 0 & 0 \\ 0.2857 & 0.2857 & 0.4286 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0714 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0714 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0714 \end{pmatrix} \quad (23)$$

## Références

- [1] *Application of Data Mining Techniques to Jet Grouting Columns Design - ProQuest.*  
URL : <https://www.proquest.com/openview/171f3c95ef0aca3ae73da8b8c725a87e/1?pq-origsite=gscholar&cbl=2026366>.
- [2] Vladimir CHERKASSKY et Yunqian MA. "Practical selection of SVM parameters and noise estimation for SVM regression". In : *Neural Networks* 1 (1<sup>er</sup> jan. 2004).
- [3] Patrick GANNE et al. "Deep Soil Mix technology in Belgium : Effect of Inclusions on Design Properties". In : (19 déc. 2012), p. 357-366. DOI : 10.1061 / 9780784412350.0021. URL : <https://ascelibrary.org/doi/10.1061/9780784412350.0021>.
- [4] Rohit GIRDHAR et al. *Learning a Predictable and Generative Vector Representation for Objects*. 31 août 2016. URL : <http://arxiv.org/abs/1603.08637>.
- [5] P HABIB et P DUFFAUT. "EFFET DE LA TENSION SUPERFICIELLE SUR LA RÉSISTANCE DES MATÉRIAUX POREUX". In : (), p. 8.
- [6] Victor LUDVIG. *0vl0/microstructure-simulation-3D*. URL : <https://github.com/0vl0/microstructure-simulation-3D>.
- [7] Victor LUDVIG. *0vl0/normalize3D*. URL : <https://github.com/0vl0/normalize3D>.

- [8] Victor LUDVIG. *0vl0/supervised-learning-UGE*. URL : <https://github.com/0vl0/supervised-learning-UGE>.
- [9] Van LYSEBETTEN. "NUMERICAL\_MODELING\_OF\_FRACTURING\_IN\_SOIL\_MIX\_MATERIAL". In : (), p. 7.
- [10] Shankar Rao PANDALA. *Lazy Predict*. original-date : 2019-11-16T09:56:35Z. 6 déc. 2022. URL : <https://github.com/shankarpandala/lazypredict> (visité le 06/12/2022).
- [11] Fabian PEDREGOSA et al. "Scikit-learn : Machine Learning in Python". In : *MACHINE LEARNING IN PYTHON* (), p. 6.
- [12] D. ROWENHORST et al. "Consistent representations of and conversions between 3D rotations". In : *Modelling and Simulation in Materials Science and Engineering* 23 (oct. 2015), p. 083501. URL : <https://dx.doi.org/10.1088/0965-0393/23/8/083501>.
- [13] Li SHEN, Hany FARID et Mark A. MCPEEK. "Modeling Three-Dimensional Morphological Structures Using Spherical Harmonics". In : *Evolution* 63.4 (2009). \_eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1558-5646.2008.00557.x>, p. 1003-1016. ISSN : 1558-5646. DOI : 10.1111/j.1558-5646.2008.00557.x. URL : <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1558-5646.2008.00557.x> (visité le 06/12/2022).
- [14] G. VAN LYSEBETTEN et al. "Discrete element modeling for the study of the effect of soft inclusions on the behavior of soil mix material". In : *Computers and Geotechnics* (jan. 2014), p. 342-351. URL : <https://linkinghub.elsevier.com/retrieve/pii/S0266352X13001596>.