Multiscale Modelling Raport No 1

Students name:	Radosław Bigaj
Field of study:	Informatyka Stosowana
Number of album:	285494
Group number:	1
Repostitory address:	https://github.com/Barcol/multiscale

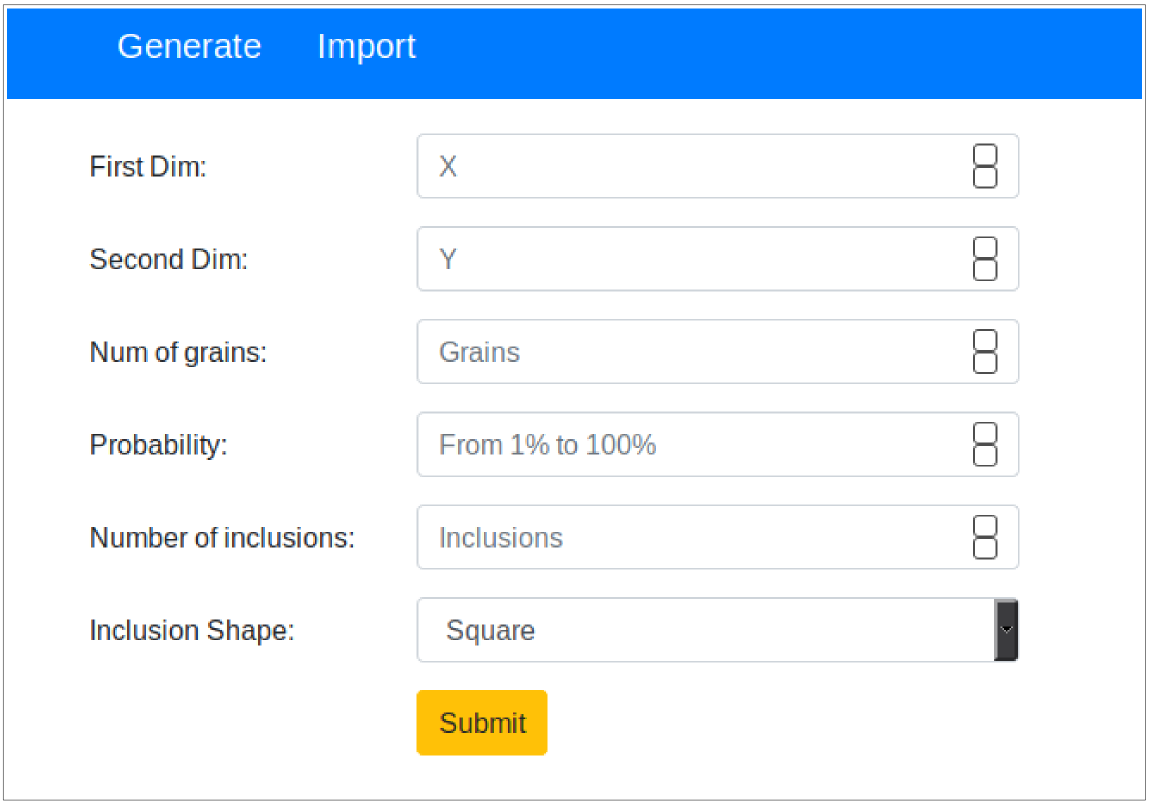
1. Used technology

Entire application was created with Python language, with Flask framework.

The GUI was created with simple HTML, along with Bootstrap 4.0.

Flask allows to use HTML templates that are connected to application core.

2. Graphical User Interface



First Dim: X

Second Dim: Y

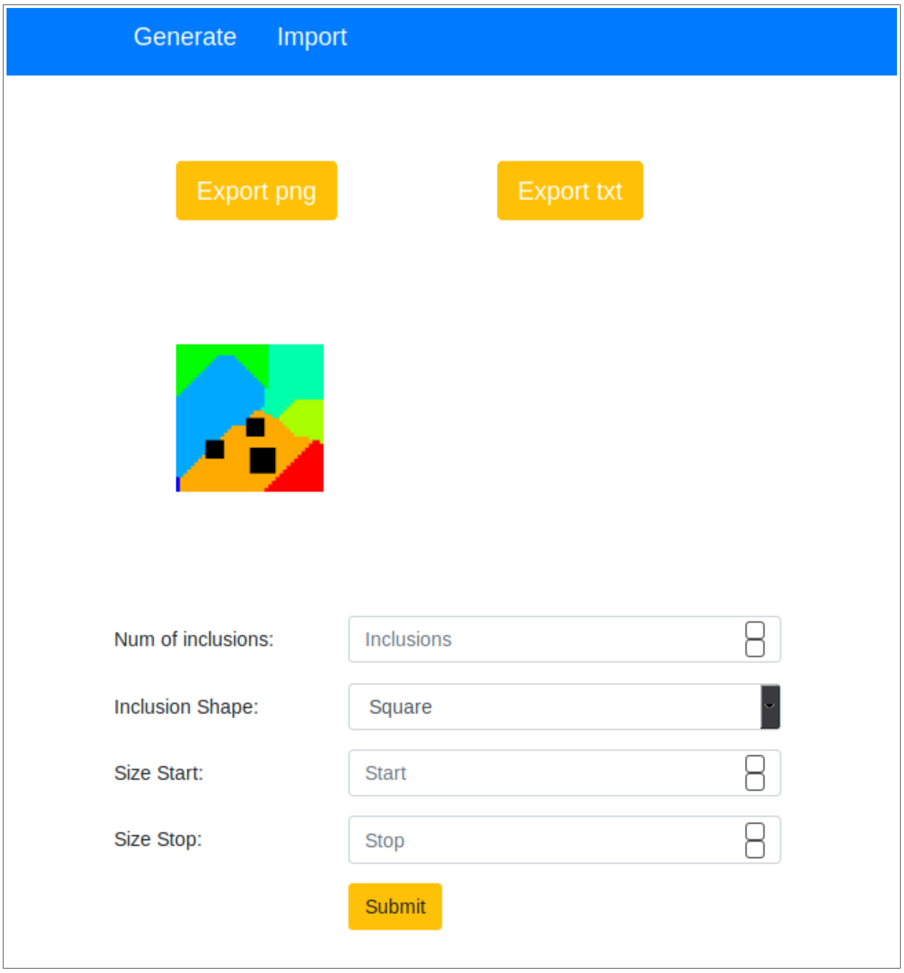
Num of grains: Grains

Probability: From 1% to 100%

Number of inclusions: Inclusions

Inclusion Shape: Square

Picture 1. Main application window.



Num of inclusions: Inclusions

Inclusion Shape: Square

Size Start: Start

Size Stop: Stop

Picture 2. Result of simple simulation.

On pictures 1 and 2 we can see two basic application windows. Picture 1 shows us the main screen, with its inputs. Picture 2 contain GIF picture presenting grain grows, options to export result, and some input fields that can be used to generate additional inclusions.

2.1. Basic usage of application

First step user have to take is picking final dimension of simulation. To do so, user have to input X value in “First Dim” field, and Y value in “Second Dim” field. Both values should be in range from 2 to 300.

Next, user should specify how many randomly located grains should be created when simulation starts. This number cannot exceed the bigger one of two dimensions (for example if the simulation is 150x250, then the maximum number of grains is 250).

Probability of a cell changing its state is determined in “Probability” field. It should vary from 1% to 100% for VonNeumann neighbeirhood method, but can also be equal to 0, for the Moore neighbeirhood method.

Last two input fields are used to specify type of inclusions. First of them can take integer arguments to determine number of inclusions. Second one is a select dropdown list that allows user to pick a “round” inclusion shape, or “square” inclusion shape. On addition, the shape can be rectangular, if dimension X is not equal to dimension Y of simulation space.

2.2. Results page

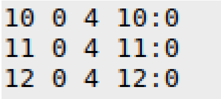
On the top of result page (shown in Picture 2) there are buttons used to export result as *.png and *.txt files.

Below them, theres the result of simulation. It is presented in a form of a GIF, and that allows us to observe growing process as it goes.

The final part of this page, can be used to rerun simulation, but with addition of more inclusions. We can pick their shape and number, as in the main page.

2.3 Import and Export

The result is presented as animated GIF, but only graphical form in can be exported into is PNG. It makes it much easier to import it.

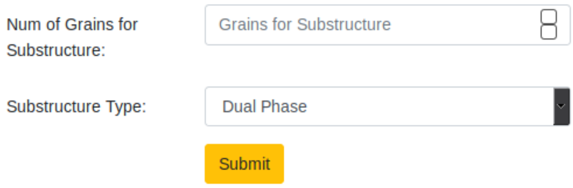


```
10 0 4 10:0
11 0 4 11:0
12 0 4 12:0
```

Picture 3. Three example lines of .txt file

The other file type that is aplicable in both import and export features, is .txt. It is a simple text format in which we specify state of each pixel at every line. Example of this can be see on Picture 3.

2.4. Dual Phase substructure



Num of Grains for Substructure: Grains for Substructure

Substructure Type: Dual Phase

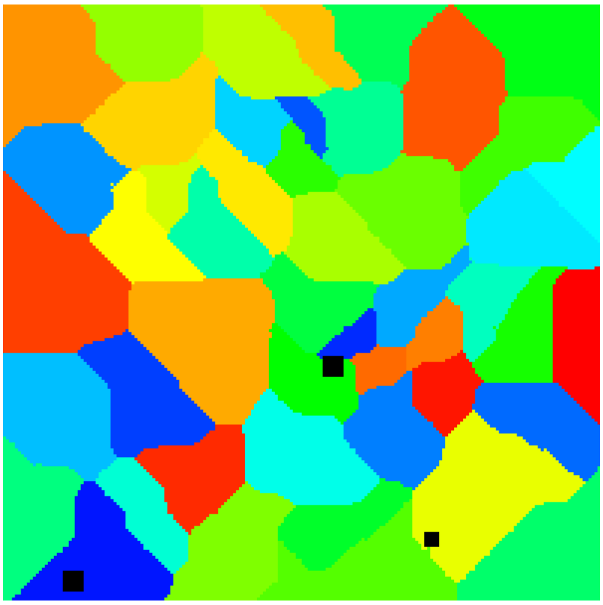
Picture 4. Dual Phase settings.

The application allows us to use Dual Phase simulation. To do so, user should head to bottom of result page, where he can find settings shown in Picture 4. We can specify number of grains, and apply it with Submit button. “Substructure Type” dropdown select list should be ignored.

3. Summary

The application can be used to generate a broad variety of real-life-like microstructures. There can be many different configurations, that lead to many different result.

Example of such simulation is shown on Picture 6.



Picture 6. Example result of advanced simulation