Multiscale Modeling Raport

Students name: Stanisław Hodurek

Field of study: Informatyka Techniczna

Number of album: 285900

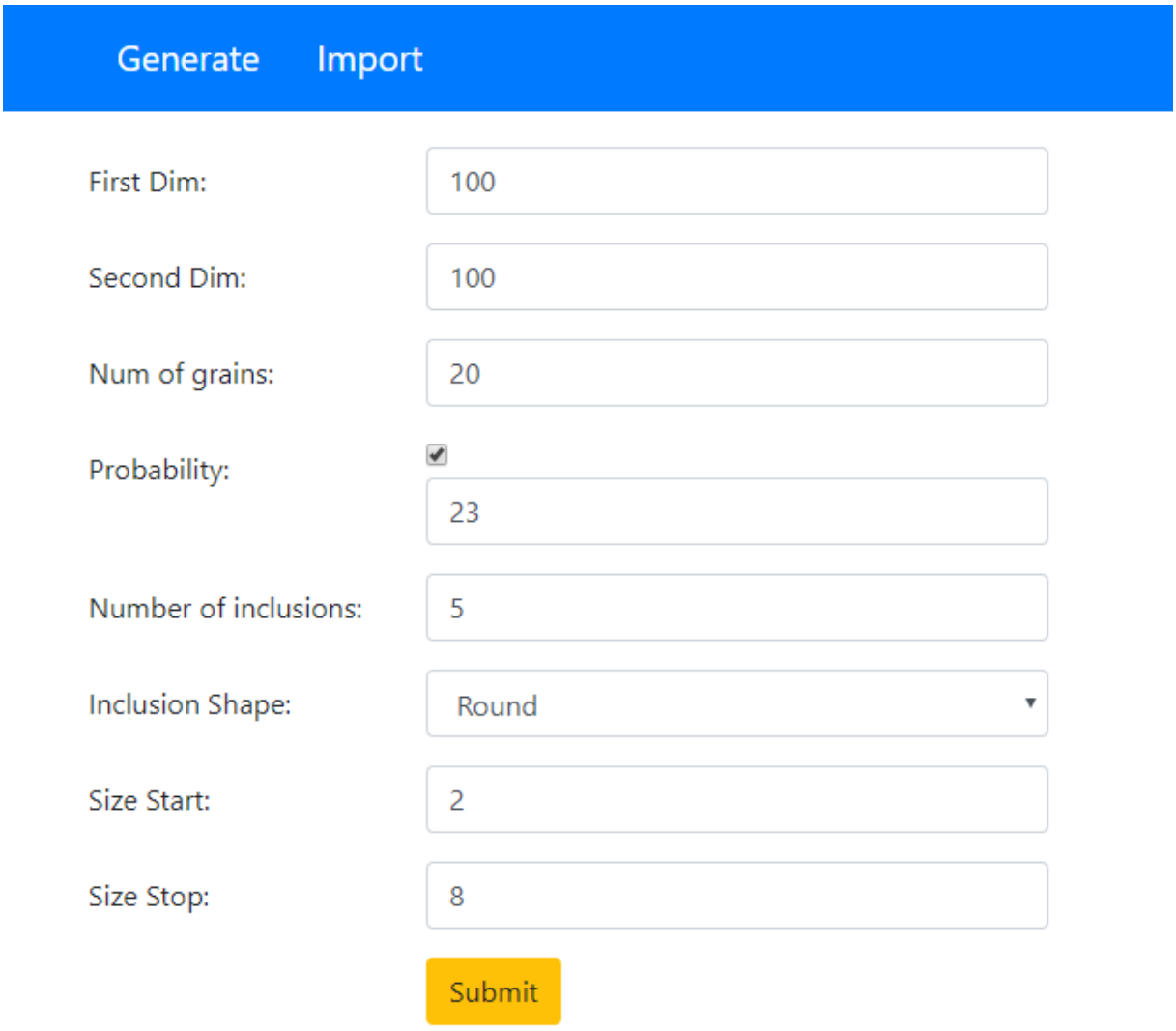
Group numer: 1

Repository address: https://github.com/staxxs3/multiscale

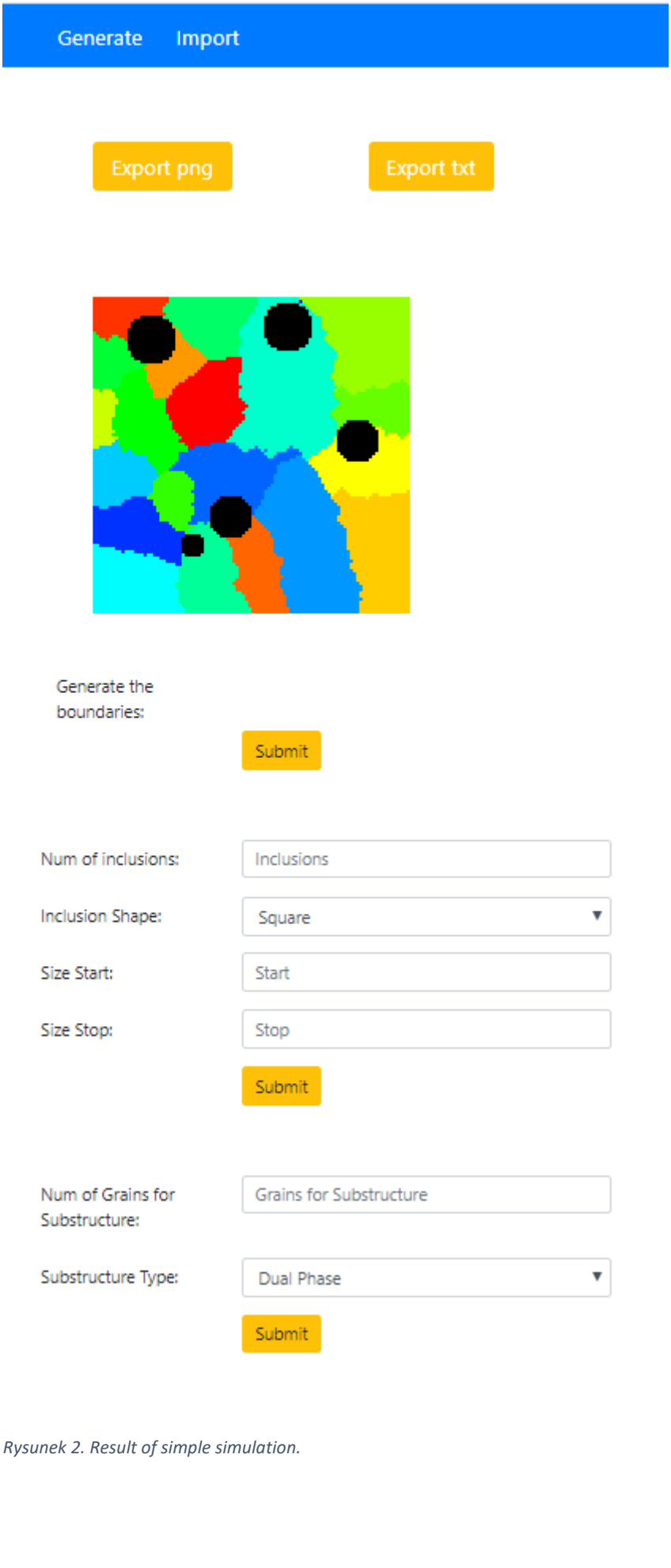
1. Used technology

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



Rysunek 1. Main application window.



Rysunek 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). 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, there’s 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.

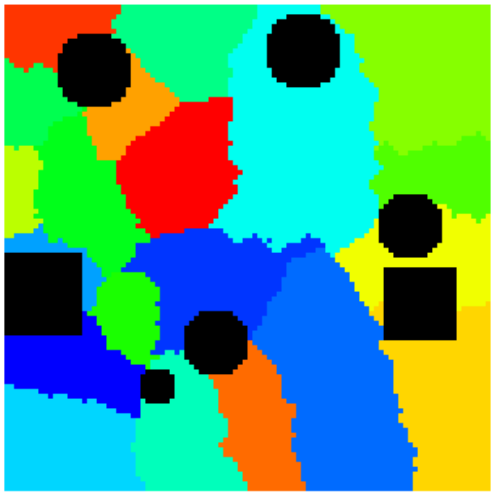
```
150 130 3
0 0 3 0:0
1 0 3 1:0
2 0 3 2:0
3 0 3 3:0
4 0 3 4:0
```

Rysunek 3. Examples line of txt file

The other file type that is applicable 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 seen on Picture 3.

2.4. Add inclusions

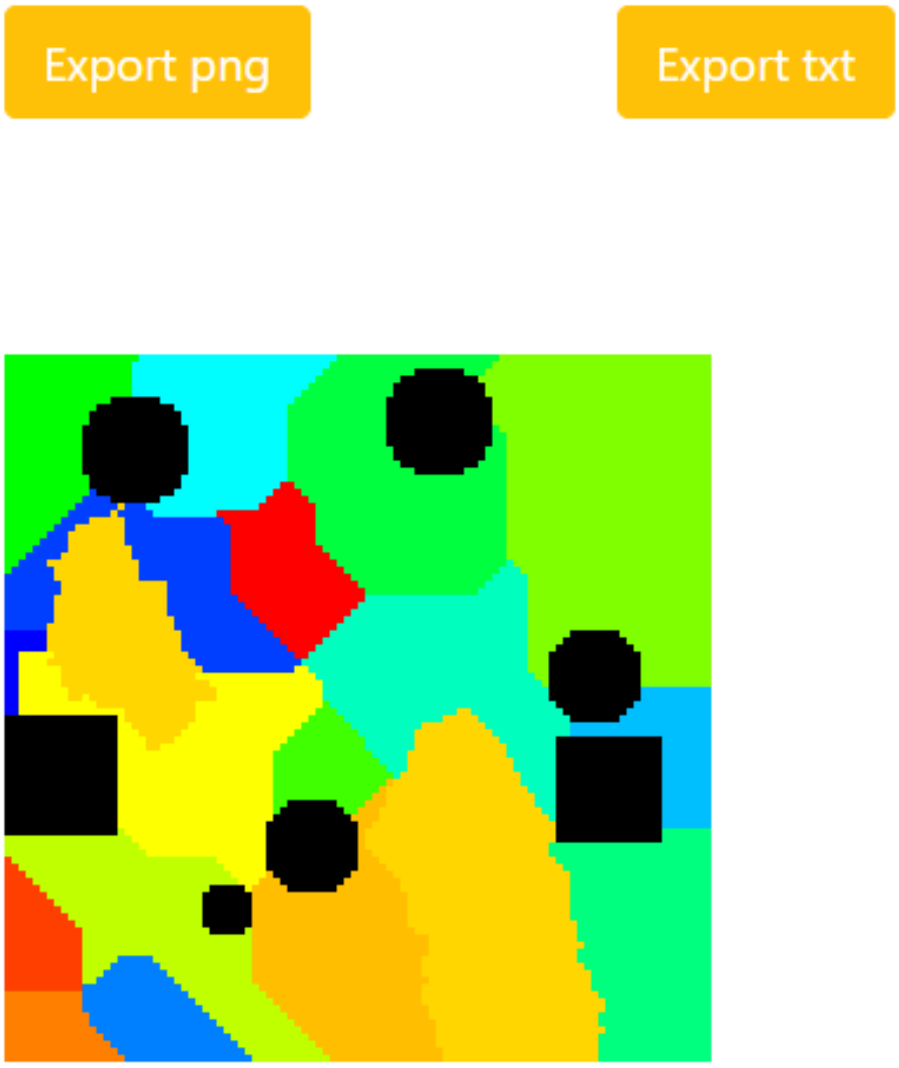
In the application in window of finish simulation we can add inclusions, on the picture I added two square inclusions.



Rysunek 4. Inclusions added

3. Dual Phase Structure

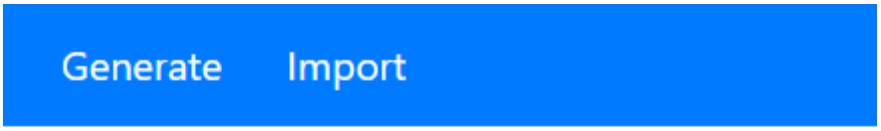
The application allows us to use Dual Phase simulation. To do so, user should head to bottom of result page, where he can find settings “Num of Grains for Substructure:” and “Substructure Type:”, when he form this properties and apply it with Submit button. He receive result like picture 5. Dual phase simulation is paint on bright orange color. Rest of grains is simulated again.

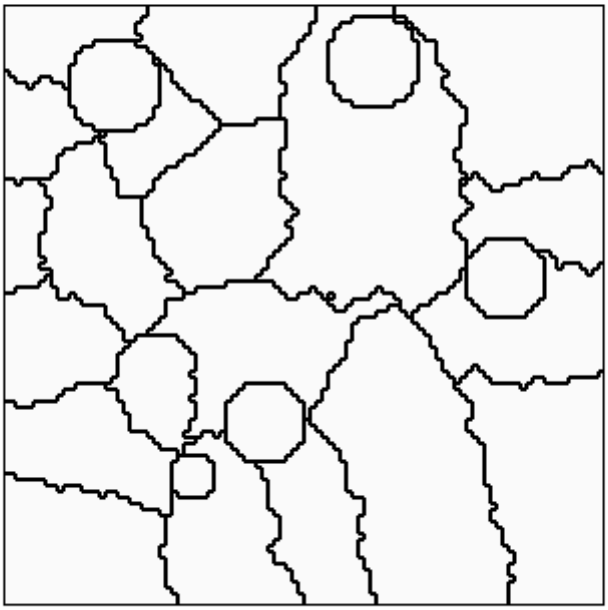


Rysunek 5. Dual-phase structure

4. Generate the boundaries

When user click on button with text “Generate the boundaries” he receive a boundaries of every grain and inclusion and saved it like picture in png format.



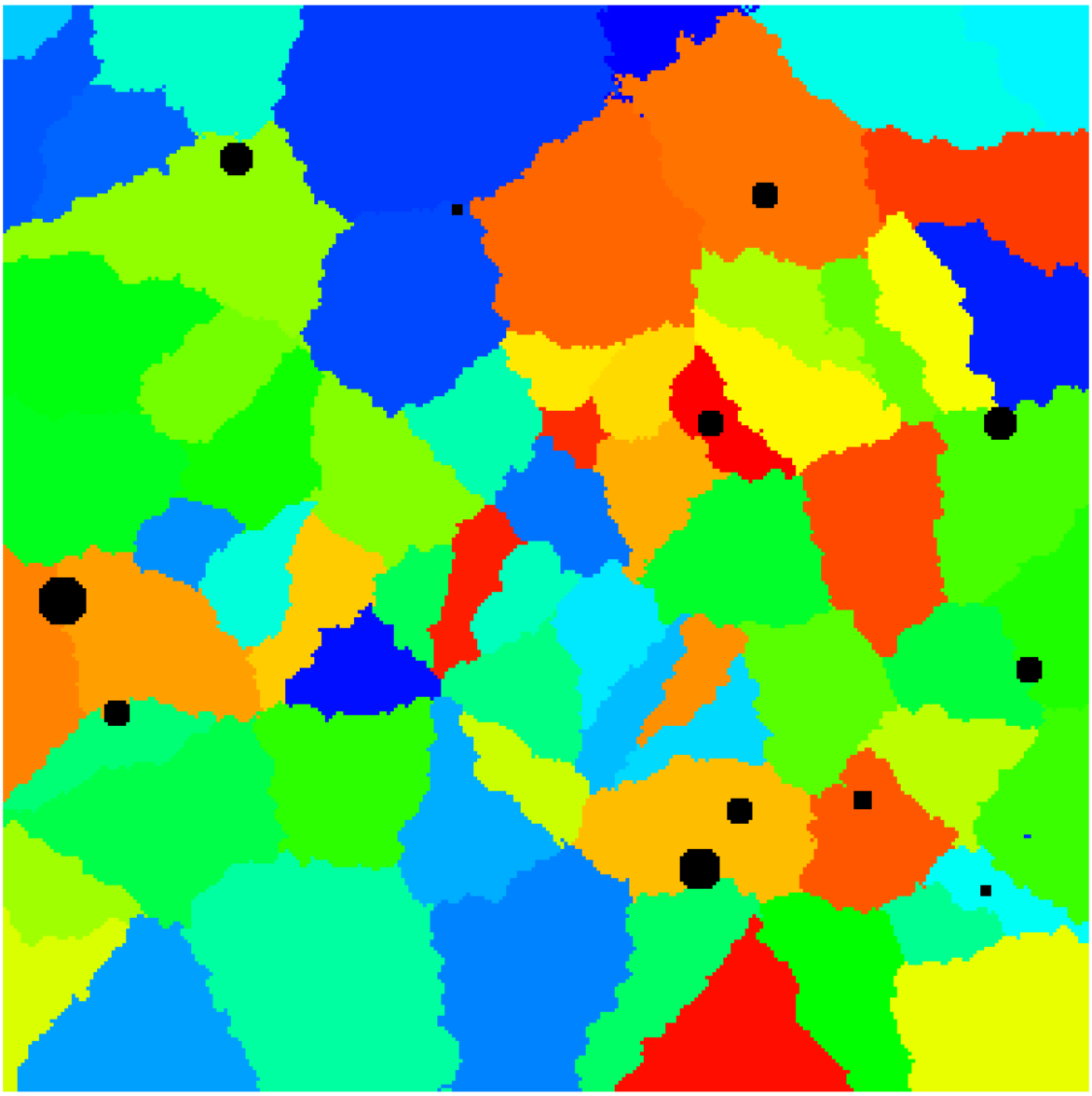


Rysunek 6. Generated boundaries

5. 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 more advanced simulation:



Rysunek 7. Advanced simulation

First dim: 300
Second dim: 300
Grains: 70
Probability 15%
Inclusions: 12 (2-8 size of inclusions)
Time of simulations: 10 minute