AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Metals Engineering and Industrial Computer Science

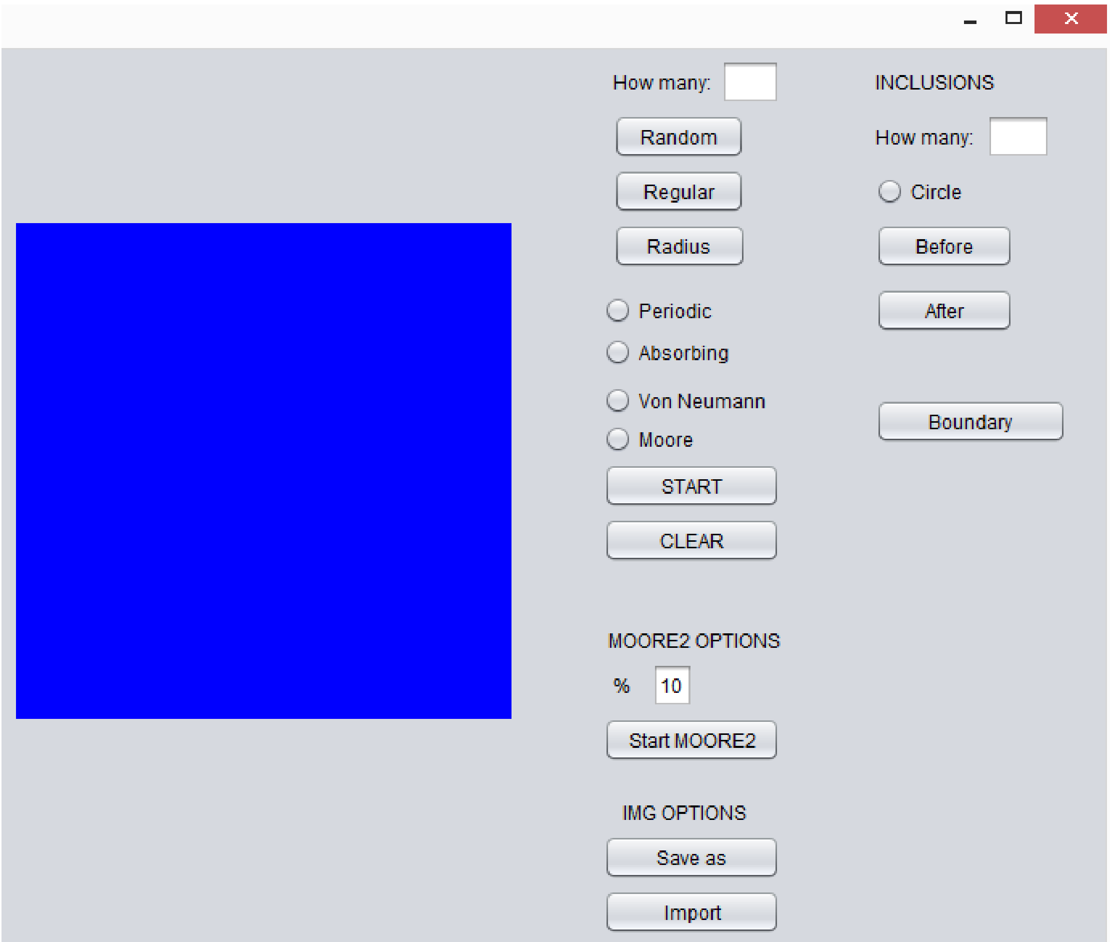
# MULTISCALE MODELLING

## "Simple grain growth with cellular automata algorithms"


**Author:** Agnieszka Bielak
**Laboratory group:** 1

**Performance date:** 08.11.2018

The aim of the project was to implement grain growth algorithms.
Simple grain growth was implemented by using Cellular Automata method. Cells have clearly defined interaction rules between each other.
Project was implemented in Java using NetBeans IDE 8.2.




Picture 1: Windows of program

The window (picture 1) is divided into two parts. The first of then is used to display simulation of grain growth. The second part is panel with options dedicated to different features.
The menu panel contains several options:

- Quanity of nucleons ;
- Boundary conditions ;
- Type of neighbourhood ;
- START/CLEAR buttons ;
- Amount of inclusion ;
- Type of inclusions ;
- BEFORE/AFTER buttons ;
- Probability for rule 4 in shape control (between 0-100) ;
- START button for rule 4 ;
- SAVE AS/IMPORT buttons for image ;
- BOUNDARY button for boundaries coloring ;

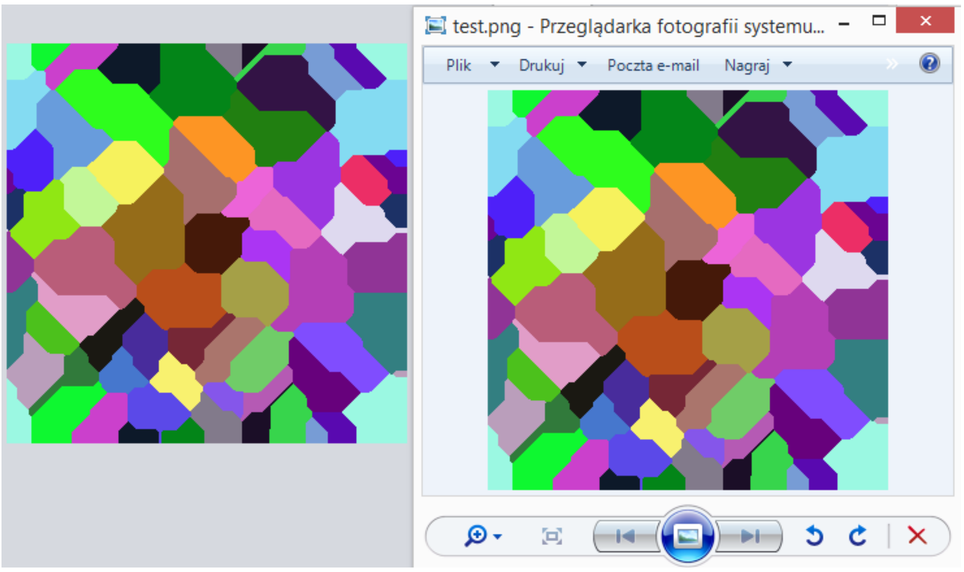
## Lab 1: simple grain growth CA + visualization

We generated and grew grains (picture 2). We can also choose number of grains. The key factor in the calculation is to define neighborhood. It affects the shape of obtained grains and hence final microstructure. User can use Moore or vonNeumann neighborhood and periodic or absorbing boundary conditions.

Picture 2: Random generated microstructure

## Lab 2: Microstructures export/import to/from txt files, pictures.

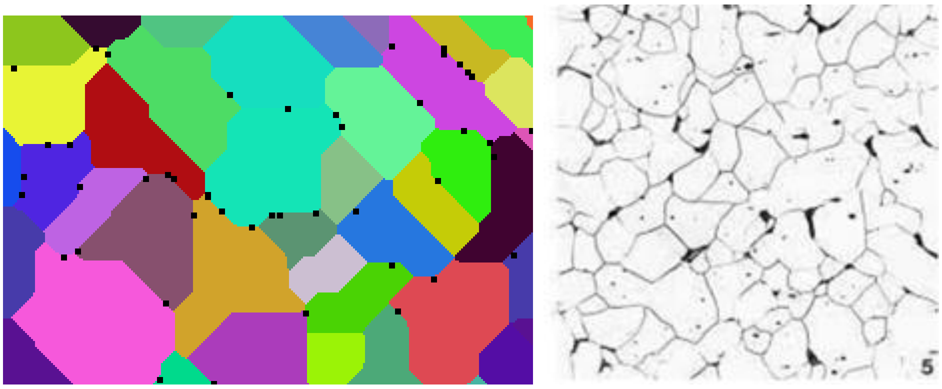
The second part of the project was to make the software available to import and export data. I used one format file: .png.



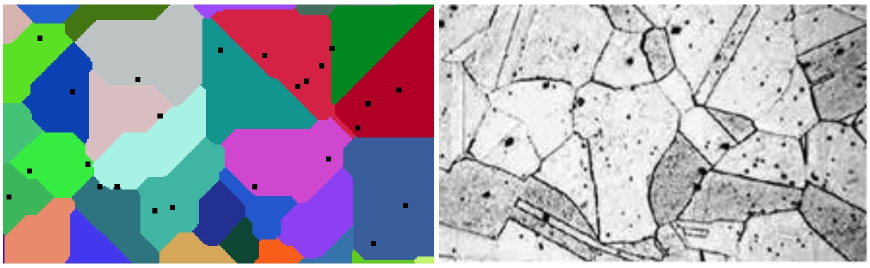

Picture 3: Example of exporting microstructure to picture

## Lab 3: Modification of cellular automata grain growth algorithm - inclusions

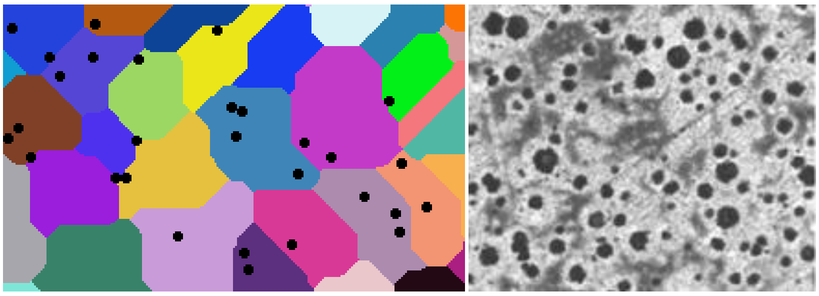
The next part of the project was to make the software available to generate inclusions. The inclusions are treated as cells that are not empty, but does not belong to the grain. Inclusions can be added Before Grain Growth and After Grain Growth and may take the form of a square or circle.



4: Microstructure from generated application with inclusion after growth and microstructure of AISI/SAE 1010



5: Microstructure from generated application and microstructure of austenitic stainless steel 00H17N14M2

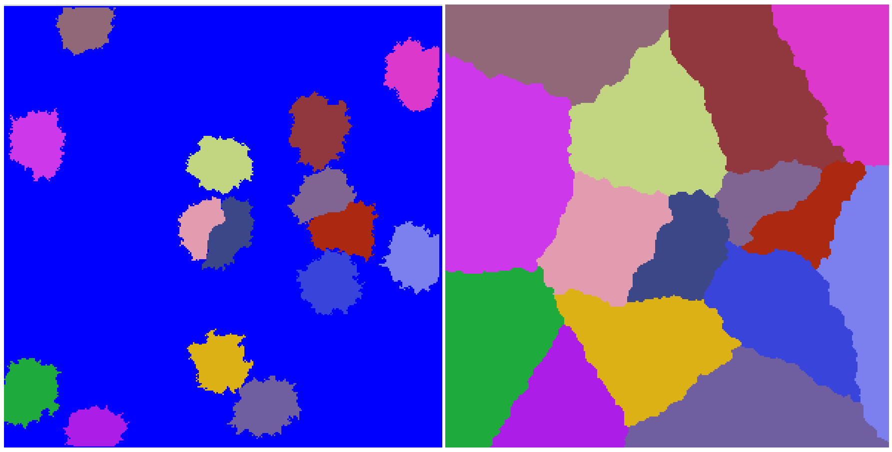


6: Microstructure from generated application with circle inclusion and microstructure of steel

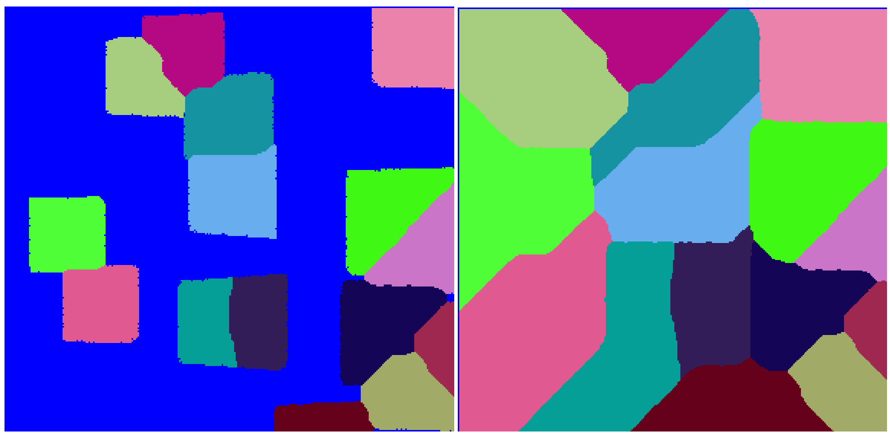
This is a comparison (picture 4, 5, 6) of real microstructure and generated by developed application. If only we properly choose a properties of our grain growth, it will be close to the real structure. This options are very useful for grain growth simulation because in practice this process is not free from any inclusions.

# Lab 4: Modification of CA grain growth algorithm - influence of grain curvature

Examples below (picture 7, 8) shows result obtained using extended Moore neighborhood. Settings for this simulation were the same but with two variants. With 10% probability (picture 6) and 90% (picture 7).



Picture 7: Extended Moore, 10% probability



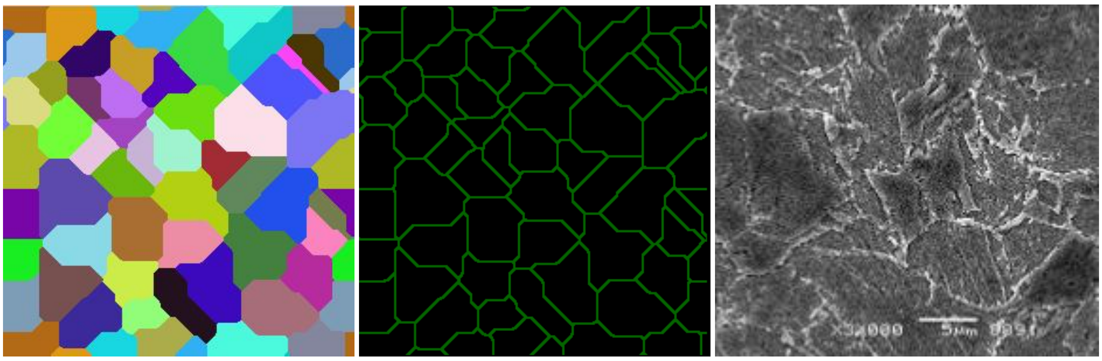
Picture 8: Extended Moore, 90% probability

# Lab 5: Modification of CA grain growth algorithm - substructures CA

Unfortunately during the implementation, I encountered difficulties in adding this functionality. As a result, it was not fully prepared.

## Lab 6: Modification of CA grain growth algorithm – boundaries coloring

The last part is the display of grain boundaries and subsequent removal.



Picture 9: Microstructure received using Moore neighbourhood and grain boundary, compared with Steel HSLA

Each microstructures have sharp grain with clear boundaries of these grains.

## Conclusion

A cellular automata is a collection of colored cells on a grid. It come in a variety of shapes and varieties depend of rules based on the states of neighboring cells. Those rules are applied iteratively for as many time steps as desired to fill the whole grid. It is possible to achieve different simulations and results by using various rules.

Thanks to CA simulation results are much more understandable as it is well visually represented. It is easier to understand grain growth, when it is possible to see simulation of it.