Multiscale Modelling - Raport

1. Introduction

The goal of this project was to implement the algorithm using cellular automata technique to divide a specific part of material into lattices of finite cells. Algorithm should simulate what happened with the grain during crystallites process in a material at the height temperature.

2. User interface

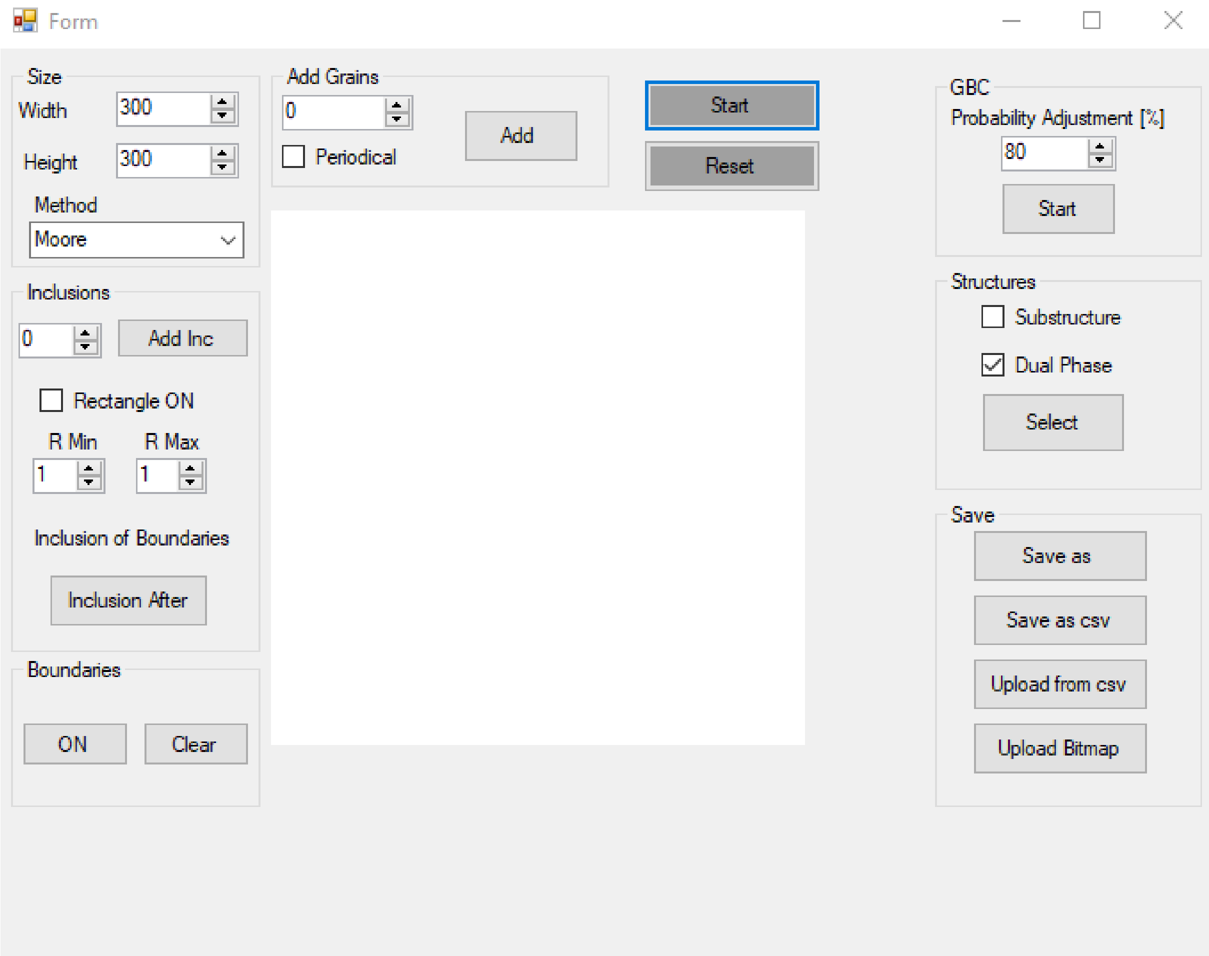

Figure 1 Main window of UI

The graphical user interface has implemented features that allows the user to customize the simulation. The main picture box is a rectangular domain of simulation. Starting with the upper left corner, user has ability to adjust the picture box where the simulation is shown, to fit to the customer requirements. Going to the right side, user can generate passed value of nucleons on the random positions on previously generated domain. Below on the left side there is an inclusion menu where user can pick how many inclusions will be added before the simulation on random position on the domain. There is possibility to choose the shape of inclusions (by default circle but checking the check box user can chose rectangle shape of the inclusions) and specify the range of the radius for circle inclusions and width of rectangle. There is also an additional option, that provide to adding inclusion after the simulation has finished, but in that case, they are placed randomly on the boundaries.

Further to right, there are two buttons which allows user to start the grain growth and reset the hole simulation. Next, we can set an algorithm of CA grain growth algorithm and set probability of shape control algorithm. On the left corner of the settings after simulation user can pick some grains preserve them in original form(substructure) or change them into uniform grain(dual-phase), either way, preventing them from growing. On the other side "Show all boundaries" button gives user a possibility to draw grains boundaries and "Clear grains" button to draw only boundaries.

User have also possibility to importing and exporting domain in .csv and .bmp format, options for that are placed in left corner.

3. Results of system implementation

For a better understanding of the discussed issue, it was decided to generate a series of the graphics where is shown the implemented functionality of the system.

3.1. Basic tests

- Number of Grains 20
- Simple grain growth algorithm
- Substructure selection

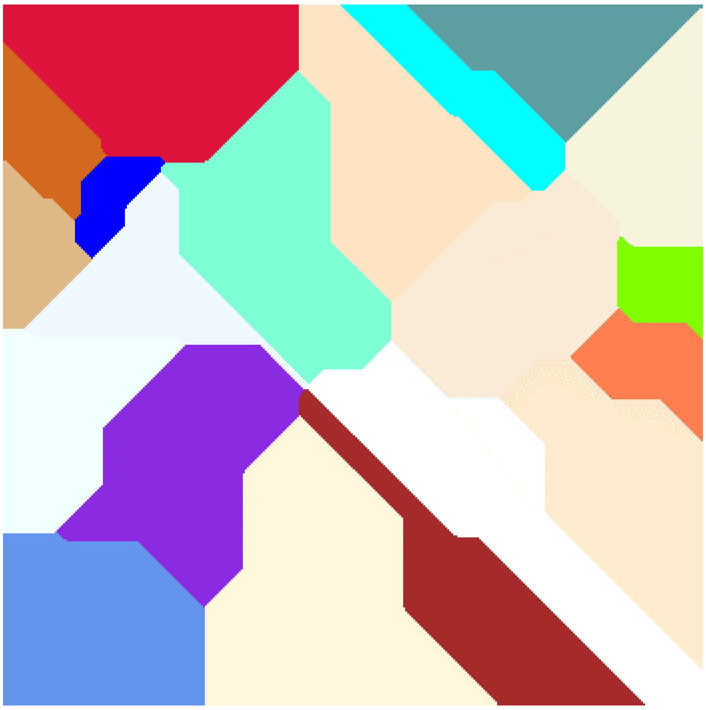

Figure 2 Domain after basic growth

- Added circle inclusions before the simulations started – number of inclusions 20

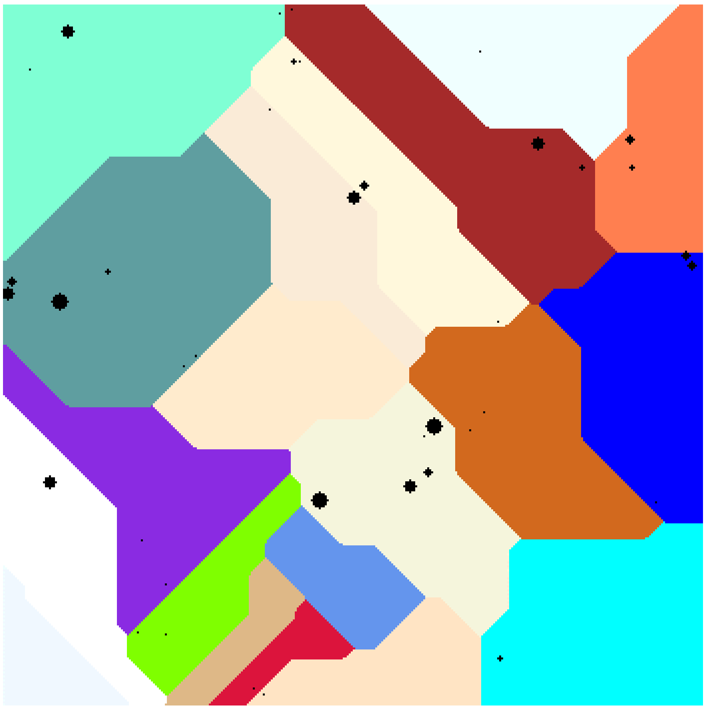

Figure 3 Circle inclusions before processing the simulation

- Added rectangle before the simulations started – number of inclusions 20

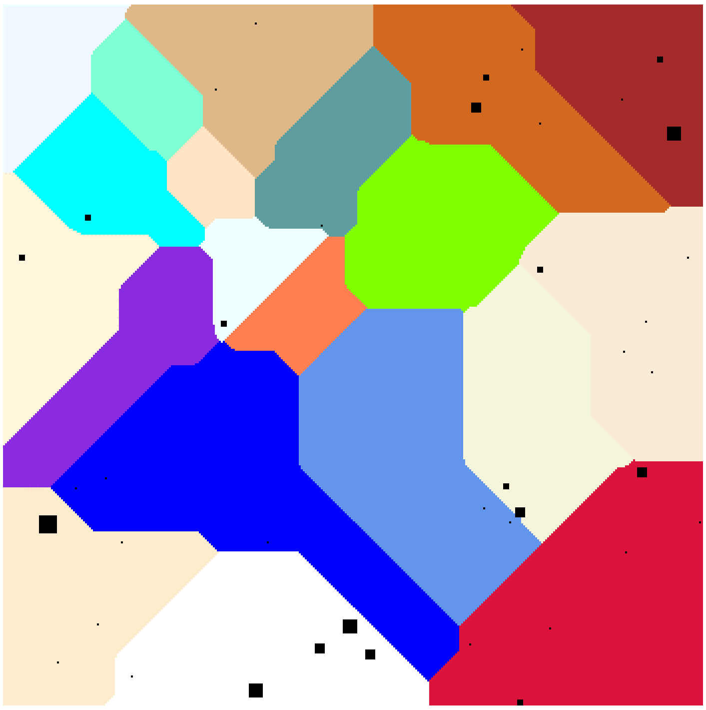

Figure 4 Rectangle inclusions before processing the simulation

- Added rectangle after the simulations started on the boundaries of grains – number of inclusions 10

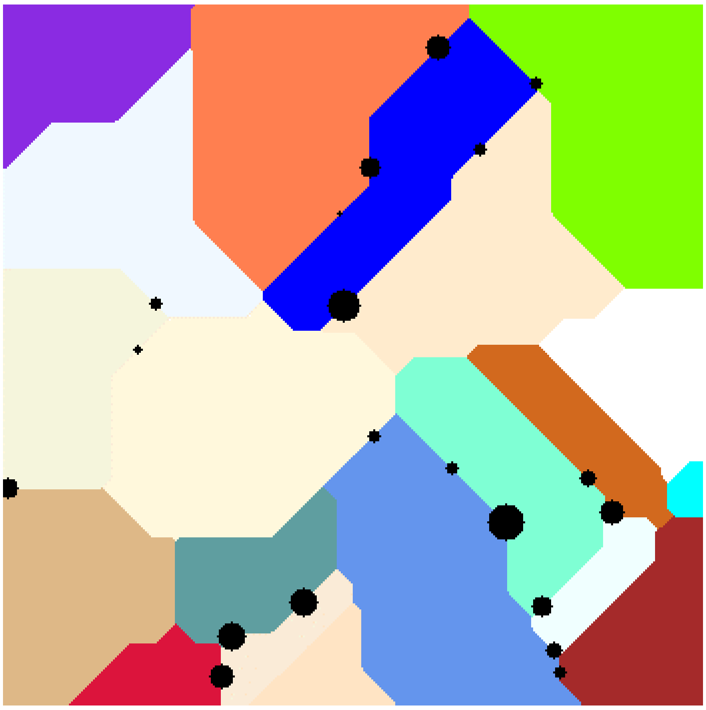

Figure 5 Circle inclusions after processing the simulation

- Added rectangle after the simulations started on the boundaries of grains – number of inclusions 10

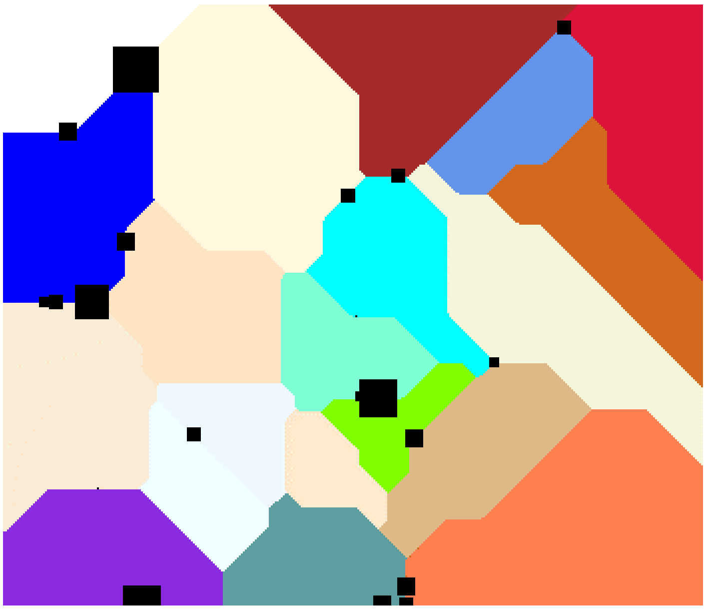

Figure 6 Rectangle inclusions after processing the simulation

3.2. Grain growth shape control

A port of implemented algorithm of grain growth was added grain boundary shape control. This functionality allows the user to adjust the probability of shape of grains. In this algorithm was implemented 4 different rules of shape control.

- Example of grain growth shape control with 10[%] probability

Figure 7 GBC 10[%] probability

- Example of grain growth shape control with 90[%] probability

Figure 8 GBC 90[%] probability

3.3. Dual phase

After simulation has finished, grains - red, green and blue were selected and merged into one grain with option „Dual Phase”. The effect of selection is combining set of three grain types into one as if they were the same.

- The Fig 9 shows the results of simple grain growth

Figure 9 Generated grains

- The Fig.10 shows three chosen grains merged into one

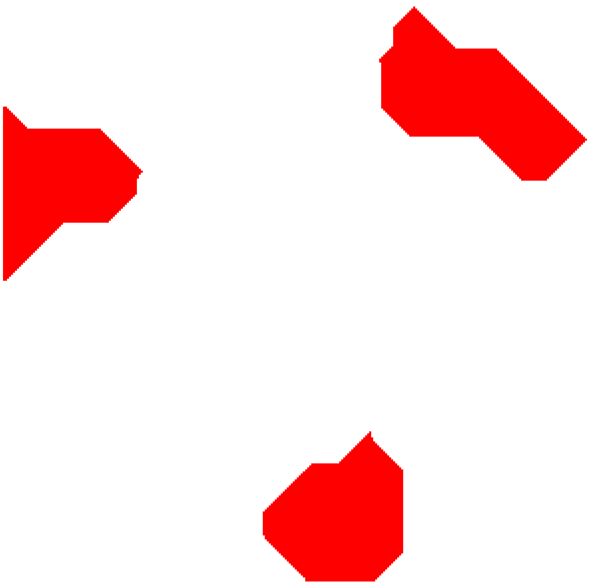

Figure 10 Domain after dual-phase selection

- The Fig.11 shows dual phase grain did not grow the rest of the domain was fill with the new grains

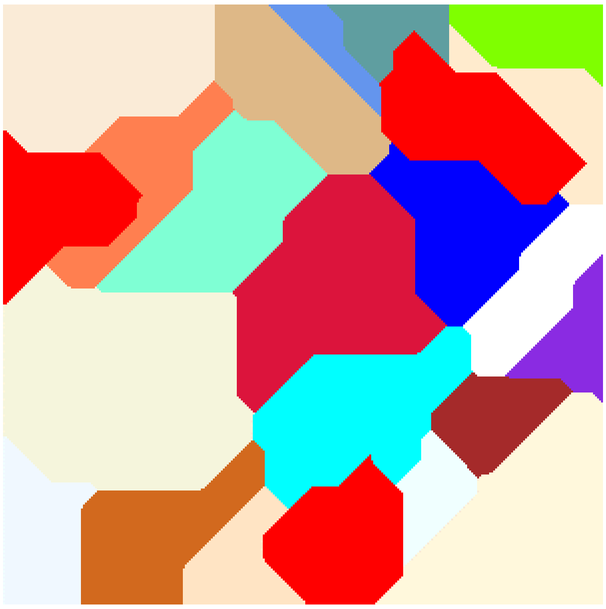

Figure 11 Domain after secondary grain growth

3.4. Boundaries

The algorithm has also boundaries detection implemented.

- After simulation boundaries detection has been added

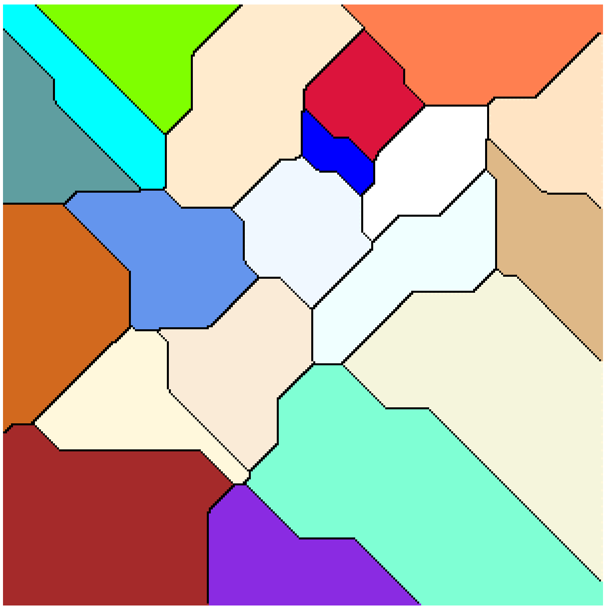

Figure 12 Grains with the selected boundaries

- The domain shows just the detected boundaries

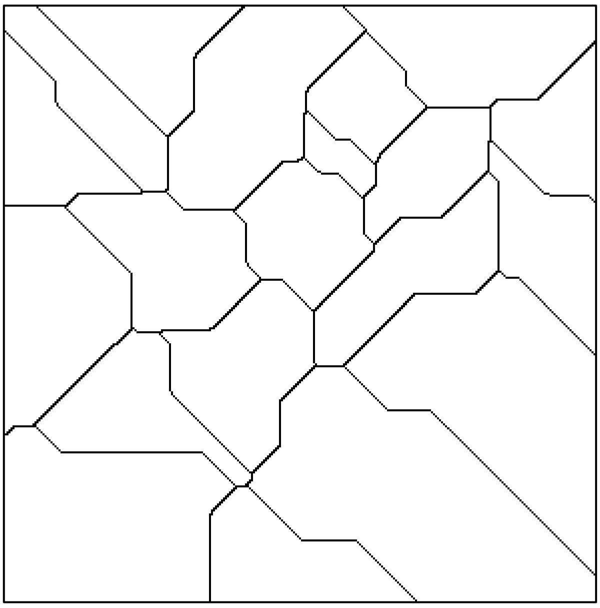

Figure 13 Domain after global grain selection

- Results after boundaries detection has been selected

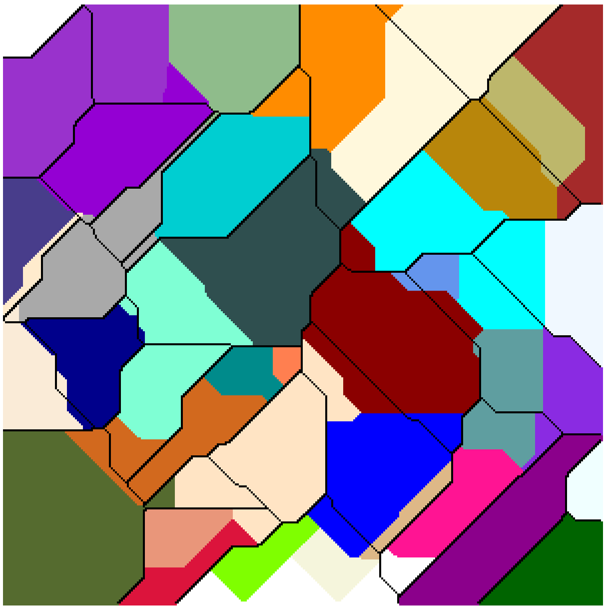

Figure 14 Domain after secondary growth

- The boundaries of selected grain

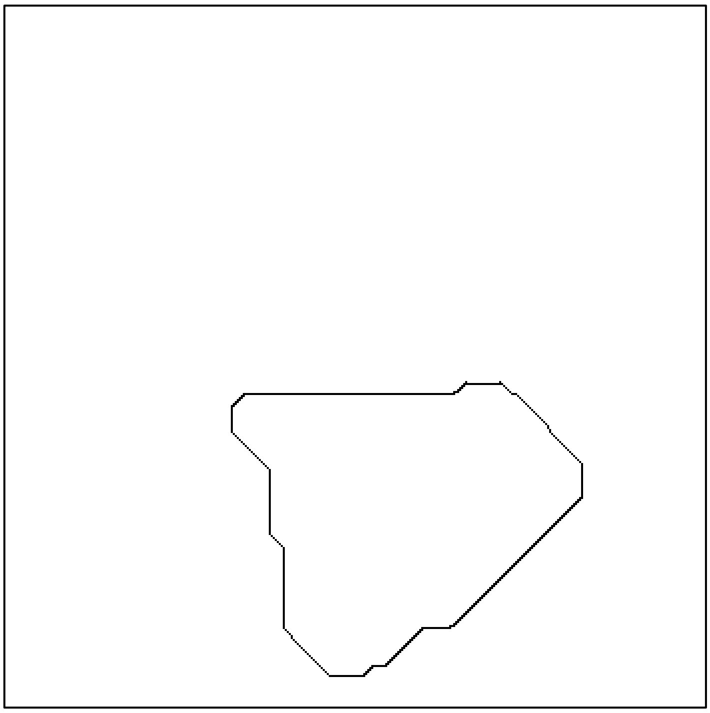

Figure 15 boundaries on selected grain

4. Summary

- From simple model of grain growth using CA there is possibility to get satisfying results, by adjusting implemented options of simulation (e.x. shape control).
- The simulation for the whole domain is reasonably quick, time complexity grows when increasing dimensions of domain. Although there is still more ways to enhance the algorithm to make it more efficient.
- The complexity of the algorithm depends on grains size. This is due to case that the smaller the grains are, the more difficult it is to represent them as round, but rather sharp edges.
- The shape control adjustment allows to get way more natural process that can remain in the higher probability the real-life examples.