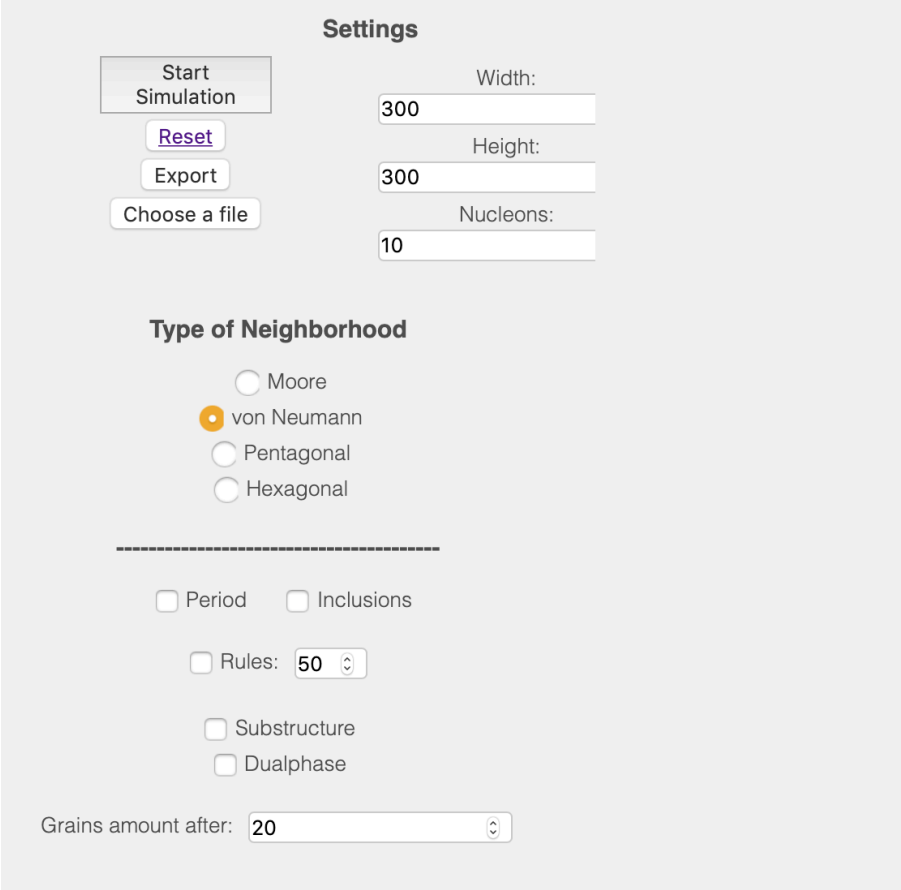
Piotr Dubiel

1st Report - Grain growth based on cellular automata

1. Introduction to algorithm and application

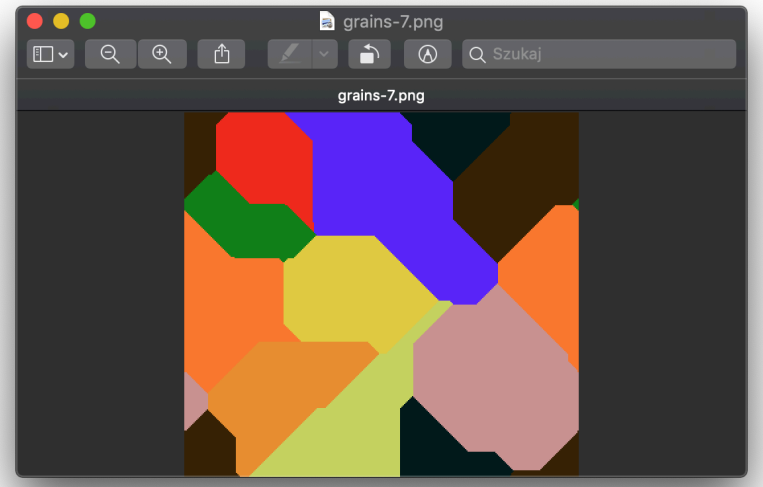
Algorithm based on cellular automata gives us a possibility to simulate the behavior of grains in metals. The program implements CA algorithm in Web environment using JavaScript. The application allows to choose the number of nucleons, the height and width of CA space, add circular or square inclusions (before and after simulation). The program has options to export finished microstructures, create substructures or dual phase and color boundaries to allow the second grain growth.



Picture 1: Graphical User Interface

2. Microstructures export/import to file

The program can export the microstructure to a bitmap. Import/Export window can be seen on Picture 1. Picture 2 depicts exported file. Also importing is possible giving the parameters of initial nucleons.

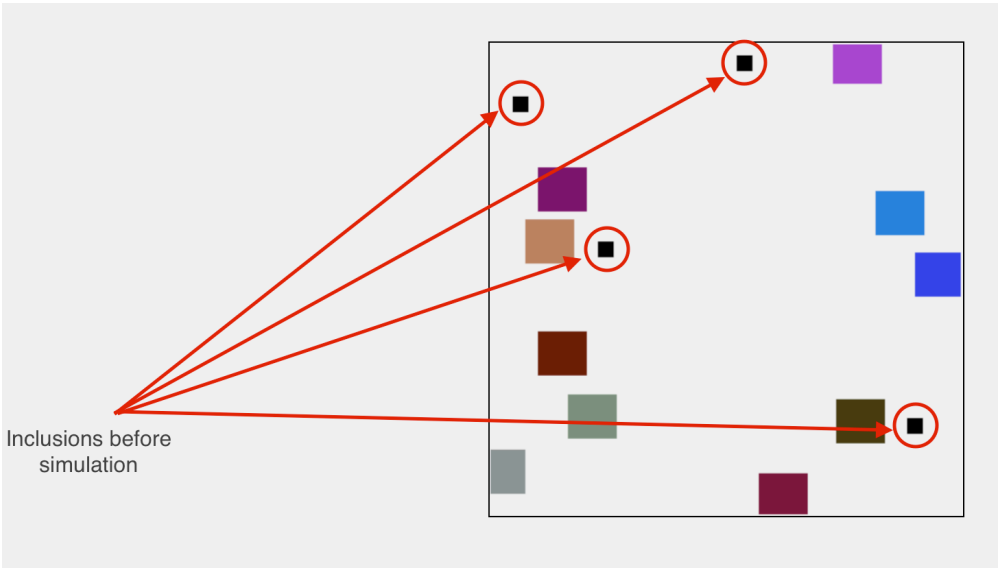


Picture 2: Exported PNG file

On the first class, we implemented application using CA algorithm for simple grains grow. The main objective was to make it work for different amount randomly placed grains, which work with Moore, von Neumann, hexagonal or pentagonal neighbourhoods. Beside the above-mentioned there are also settings for size of space and boundary conditions. Default is adsorbent boundary conditions, but user can also set a period boundary conditions.

3. Inclusions

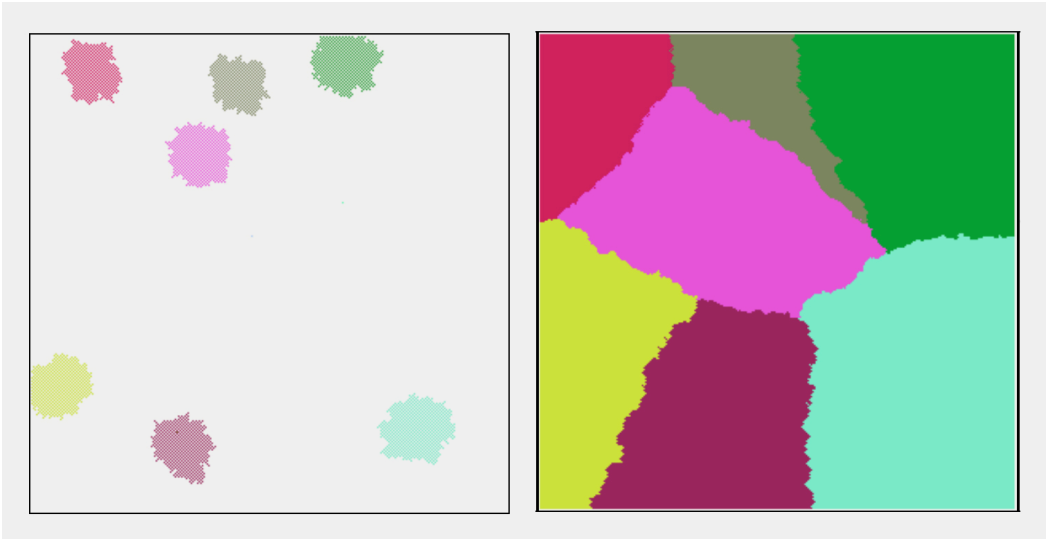
On the second class, we improved our application by adding inclusions option. We can add circular or squared inclusions before or after simulation. The default size value of the inclusion is 10px. By ticking the inclusion checkbox user can set also amount of inclusions.



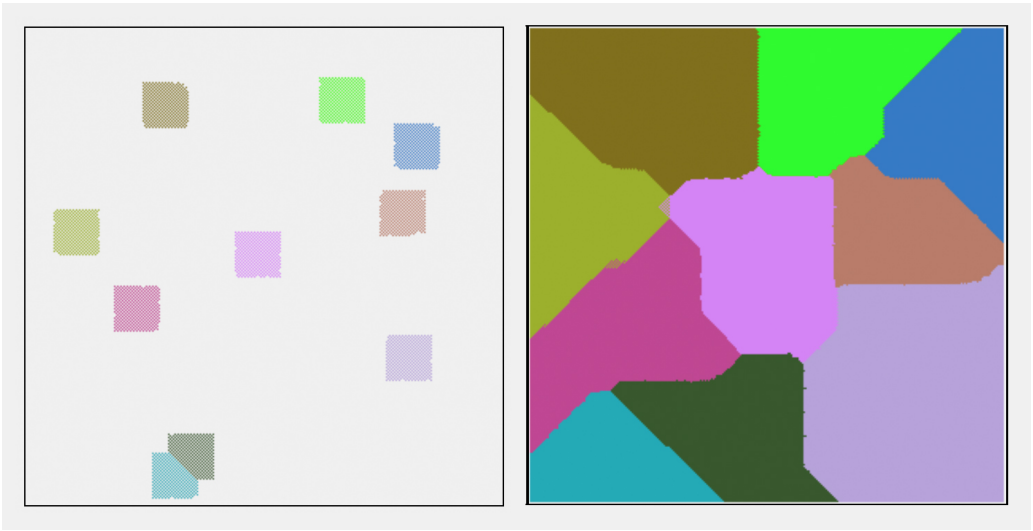
Picture 3: Square inclusions before simulation

4. More advanced grain growth algorithm based on different neighborhood types

The algorithm implemented in application changes the behavior of growth in relation to amount of already occupied points in neighbor of specific pixel. Firstly, we check if five to eight of cells neighbour's IDs has the same id (Rule 1). If it's true we change ID of our cell to that ID else we're checking the closest neighbourhood and the same happened, but now we are checking whether is at least three same neighbours (Rule 2). If it's true we're setting our cell ID to that id else we're checking further neighbours also if three of them has same ID we're setting this ID to our cell (Rule 3) else the ID of our cell depends on its all neighbours, and has X % probability chance to change (Rule 4). This probability might be set by user in the menu on the left.



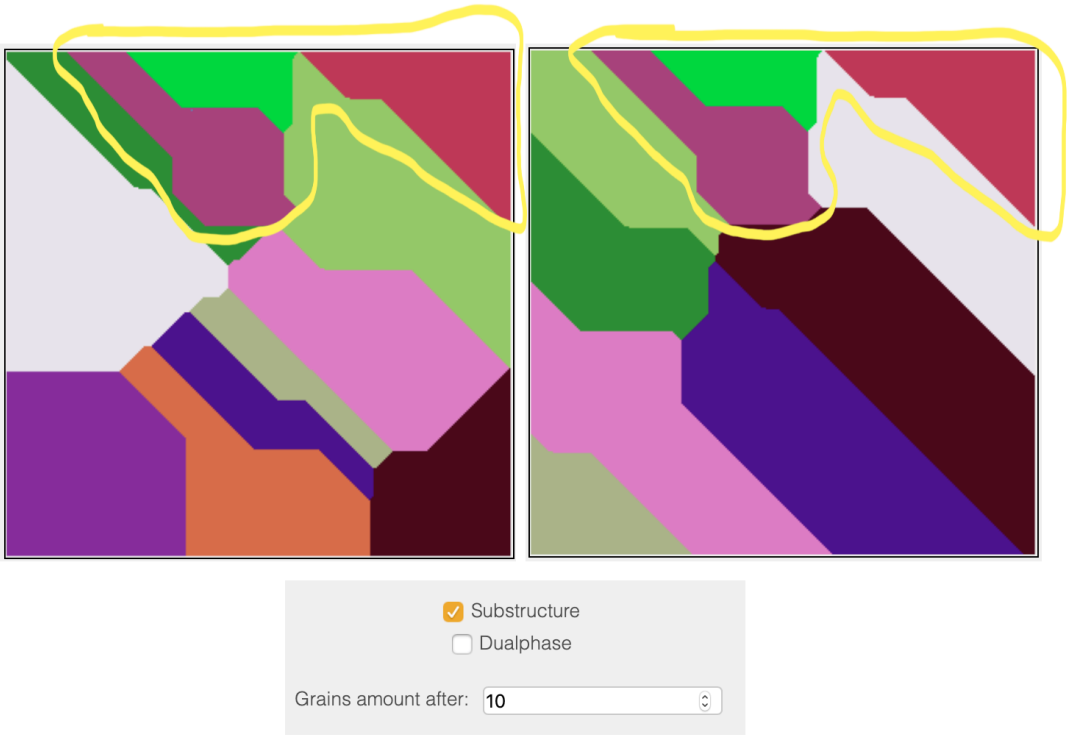
Picture 4: Example of grain growth with 20% probability for rule 4



Picture 5: Example of grain growth with 95% probability for rule 4

5. Substructure and dual phase generation

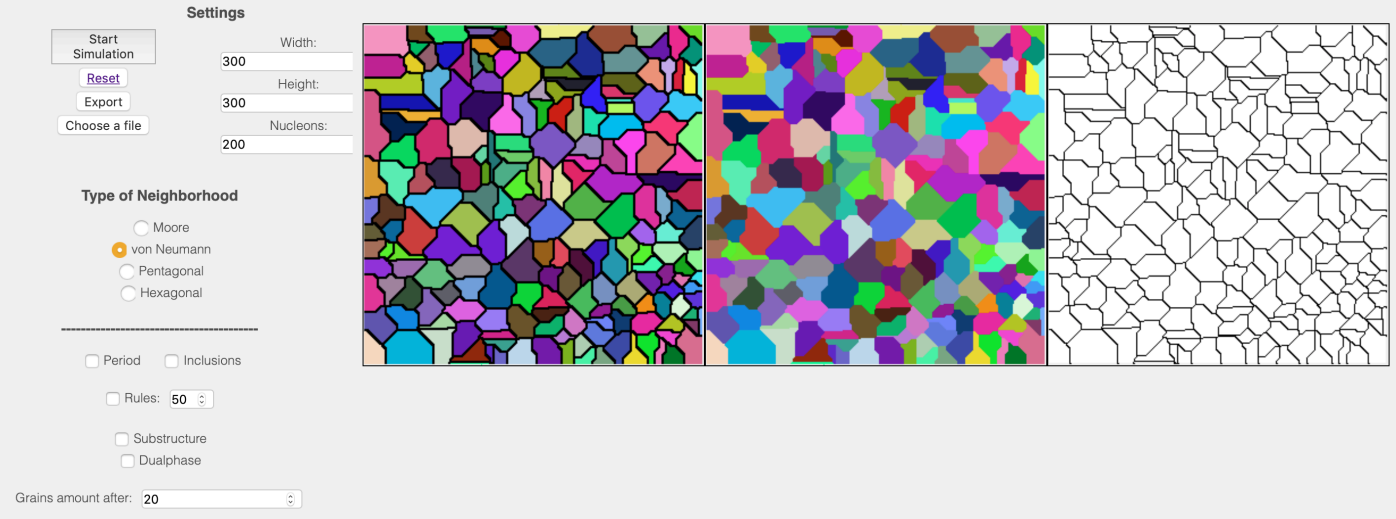
After first simulation couple of grains can be saved and, after clearing the rest, used as substructure in the next simulation. There are two types coded in application: Substructures, when the colors of substructures are preserved and the dual phase, when the color of all of them is changed to specific ID of second phase. In the simulation after substructures extraction they are not growing.



Picture 6: The example of Substructure – selected grains do not grow on the second time

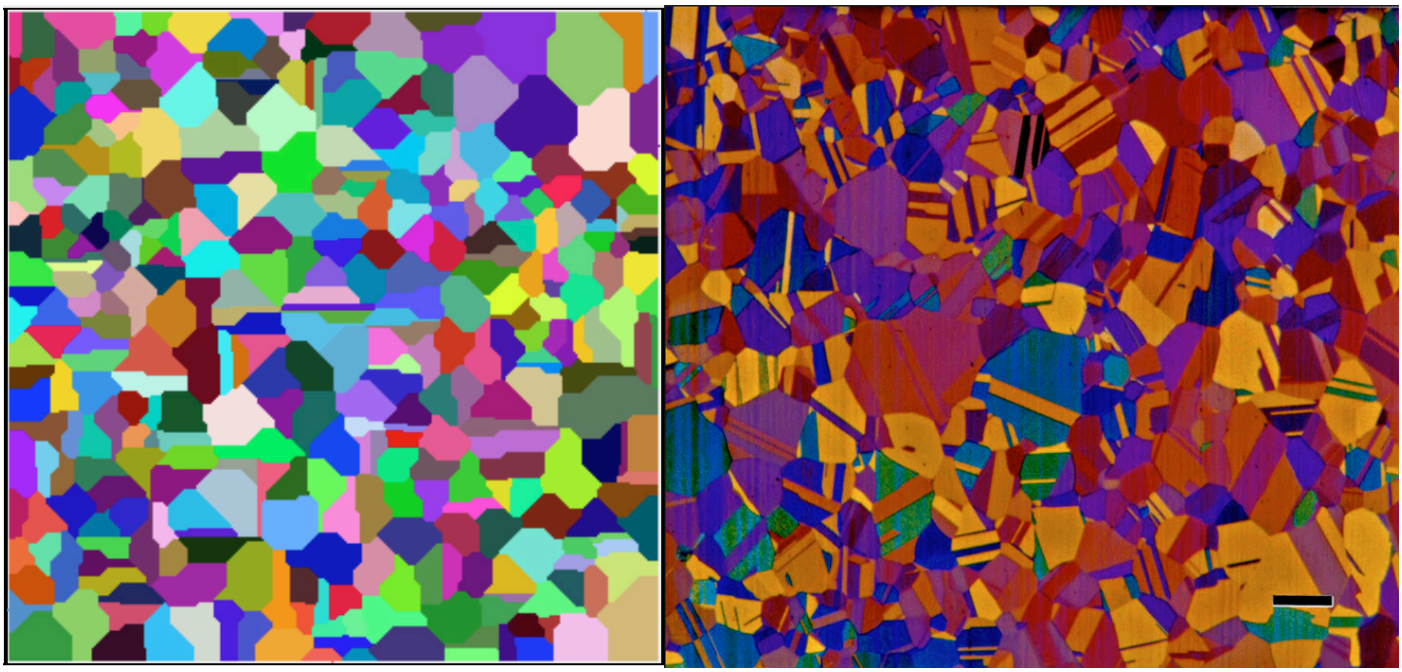
Picture 7: The example of Dual Phase – selected grains do not grow on the second time

6. Boundaries coloring



Picture 8: Boundaries coloring after simulation

7. Summary



Picture 9: Comparison of model generated in the application with a real microstructure

The simulation is providing easy and fast possibility to generate microstructure. As we can see, the microstructure generated from simulation is close to real microstructures and the cellular automata technique is giving good results when imitating the reality.

Using described above methods we can simulate grains grow, how each cell will behave when its grow and how will look whole microstructure.

Primitive grain grow is very fast and easy to implement but the results are always similar, and are not realistic. The shape of the grains is too much geometric and borders are usually straight lines and simple mathematical figures. The way to improve results was control shapes method.

Implemented inclusions provides more realistic approach to the microstructure of the material. In real life the grains have a lot of inclusions and distortions.