AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Inżynierii Metali i Informatyki Przemysłowej

Subject: Multiscale modelling

„Grain growth simulation report”

Author:
Jakub Miśkowiec
IS WIMiIP

Table of Contents

1. Technology

Application is written in Java Language. Swing library is an official Java GUI tool kit, It is used to create graphical user interface with Java. This includes basic elements such as windows, buttons, scrollbars, file choosers and combo boxes. In a project applied NetBeans IDE (it is open source Integrated Development Environment software), because It has an easy to use Swing GUI design tool to build user interfaces through dragging and dropping components, such as Buttons and Textboxes.

If you want run application on your computer you need The Java SE Development Kit (JDK) 8, which is required to install NetBeans IDE 8.2.

2. GUI Overview

Figure 1 shows graphical user interface application with the description of functions used in.

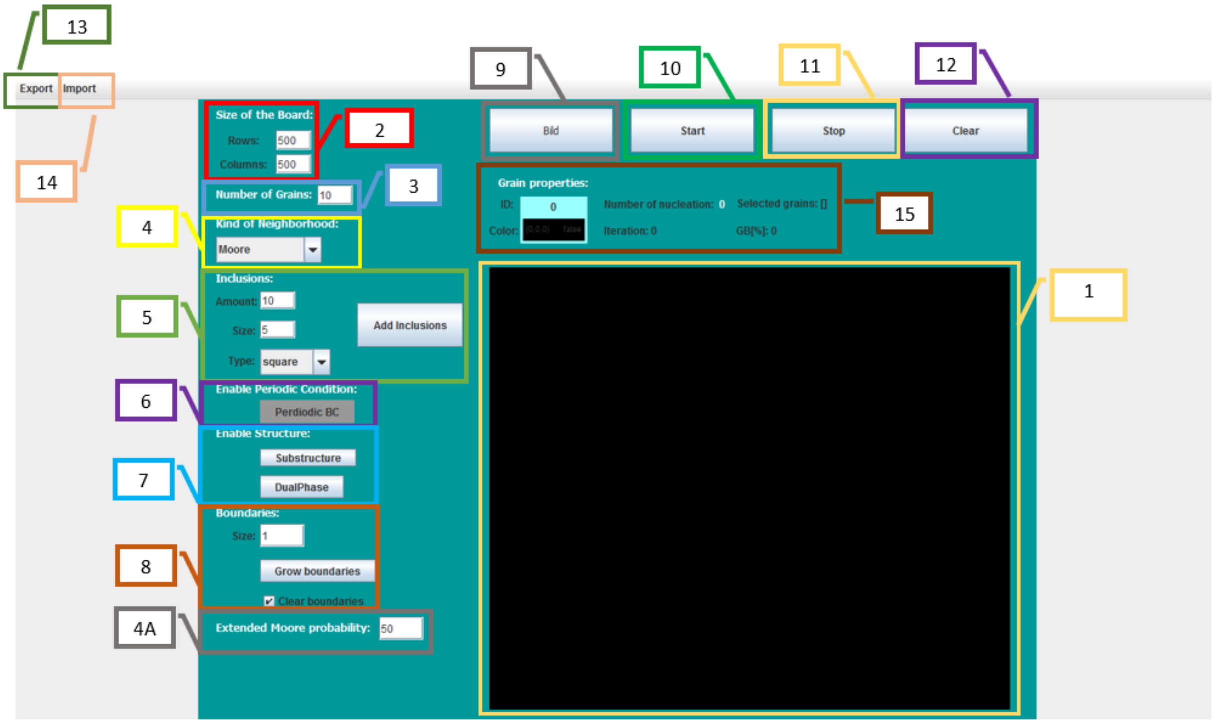

Figure 1. GUI

1) Canvas – it is graphic area, which visualizing nucleations grain growth process.

2) Size of the Board – User can change default size of area by filed Rows and Columns (width and height in pixel).

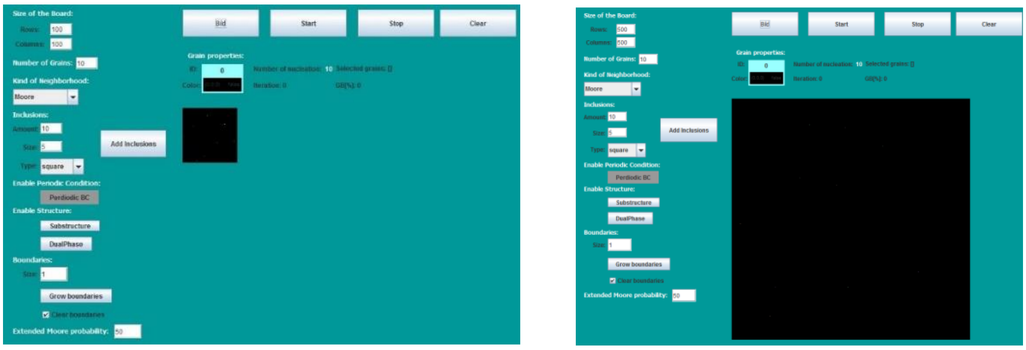

Figure 2. Board size settings (on the left 100x100, on the right 500x500)

3) Number of grains – User sets initial amount of nucleons.

4) Kind of Neighborhood – User can select one of three available options type of a neighborhood (Moore, Von Neumann, Extended Moore). If we choose Extended Moore, we have to fill Extended Moore probability field (ref. 4A).
   - Moore

Figure 3. Result of simulation using Moore’s neigborhood

   - Von Neumann

Figure 3. Result of simulation using von Neumann’s neigborhood

- Extended Moore

Based on figure 4 we can see that the boundaries are more smooth for the bigger value of probability.

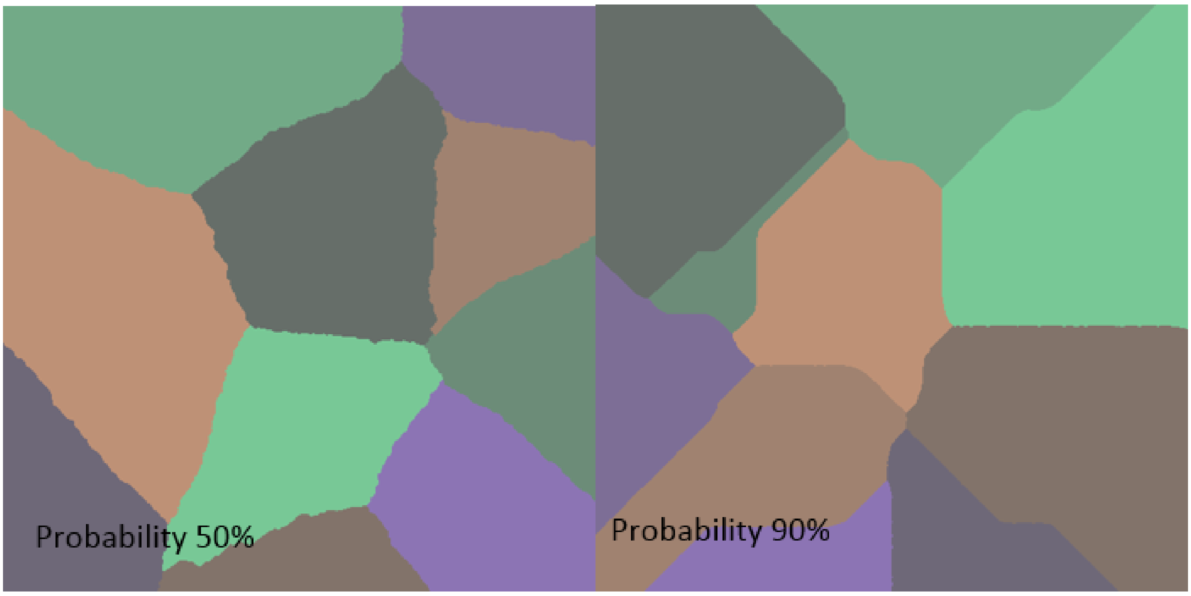

Figure 4. Result of simulation using Extended Moore's neighborhood

5) Inclusions – User can add inclusion at the start (figure 5) or end (figure 6) of simulation through pressing the button "Add Inclusions". Also User have to set properties for inclusions, that will be generated.

- amount – number of inclusions generate on the board
- size – user specifies the thickness of inclusion
- Type – user can select type of shape of inclusions shape (square, circular)

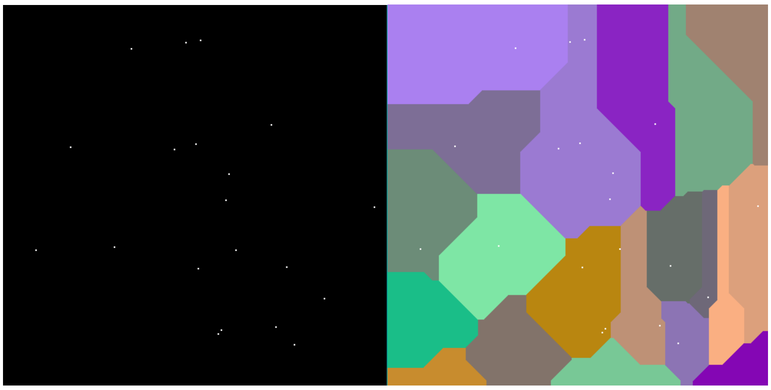

Figure 5. Added inclusion before simulation

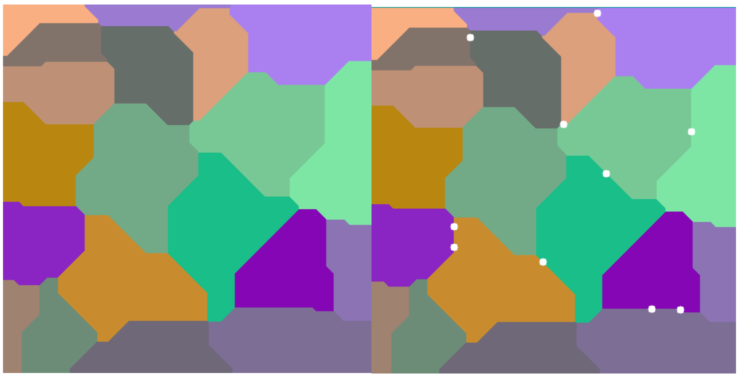

Figure 6. Added inclusion after simulation

6) Enable Periodic Condition – User can set boundary conditions. If the button is disabled (grey color) grains permeate through the board’s boundaries.

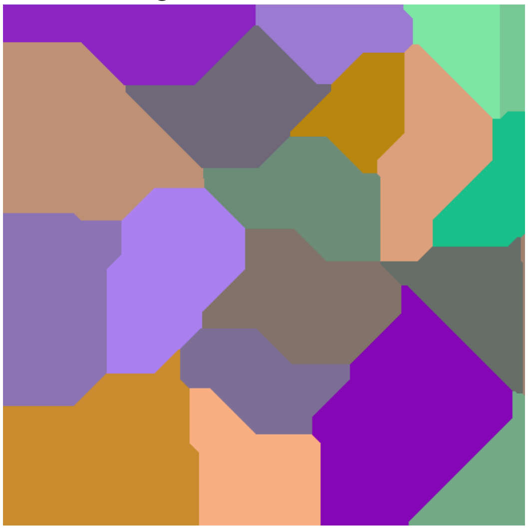

Figure 7.Grain growth simulation with absorbing boundaries conditions– disabled button

Figure 8.Grain growth simulation with periodic boundaries conditions– enabled button

7) Enable Structure – User has two options available:

- Substructure - Grains selected by the User remain after cleaning the board. The next growth will take place in a free area without seeds

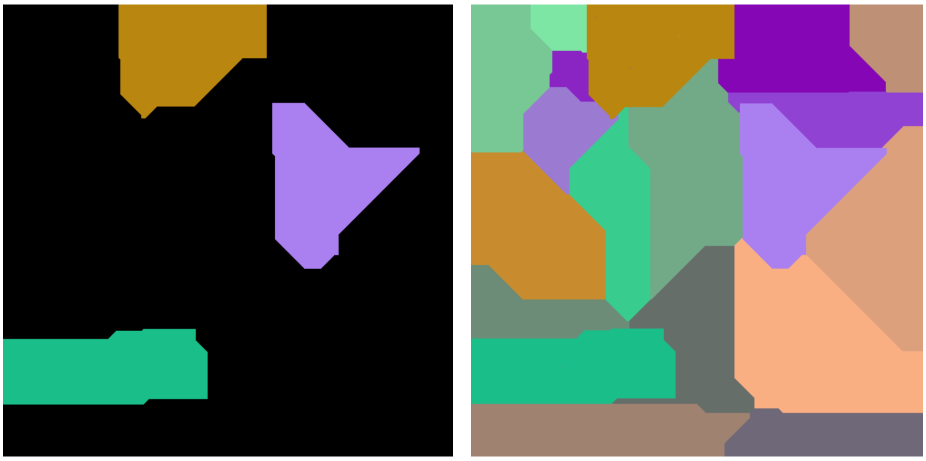

Figure 9. The result of the simulation using the substructure option

- DualPhase – Grain selected by the User change the phase and remain after cleaning the board. The next growth will take place in area without seeds, which change the phase.

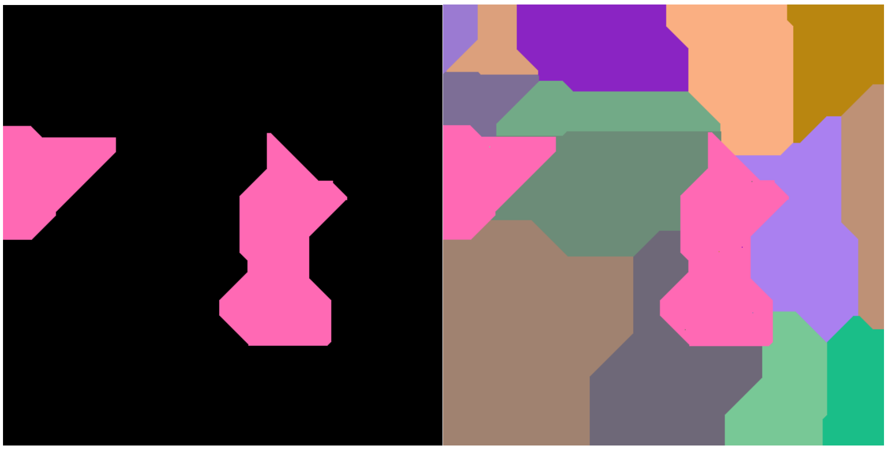

Figure 10. The result of the simulation using the dualPhase option

8) Boundaries - The user can decide that grain boundaries will appear. In addition, it can specify the size of these boundaries. If the option “Clear boundaries” is untick, only the bordes will be remain after pressing the button clear.

Figure 11. Created a microstructure after simulation

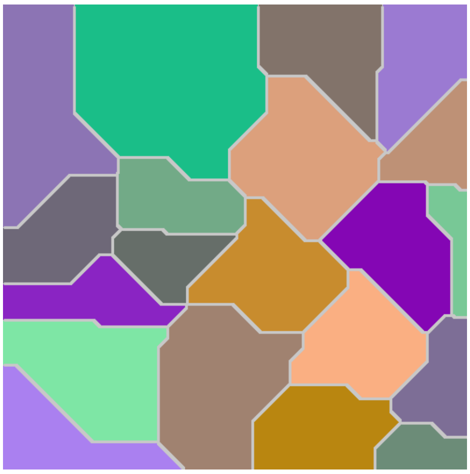

Figure 11. Layer boundarie on microstructure

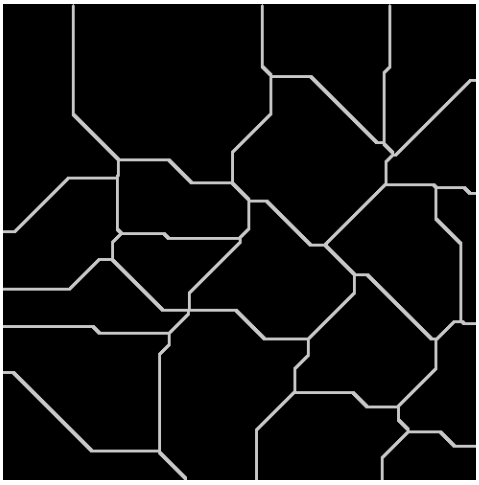

Figure 12. Framework of microstructure

9) Bild – This button generate new board with nucleons

10) Start – This button start/continue grain growth simulation

11) Stop - This button pause grain growth simulation

12) Clear - This button clear all board. The Exception is the option “Clear boundaries”, which is described in point 8.

13) Export – This option enable User to saved created microstructure to bitmap or text file. In addition, if the option “show grain borders” is tick, also boundaries will be save.

14) Import – This option enable User generate microstructure from bitmap or text file to application.

15) Grain properties – It is part of window application, which shows information about ID and color of grains, number of nucleation, amount of iterations, selected grains by the User or percentage of the board's area occupied by boundaries.

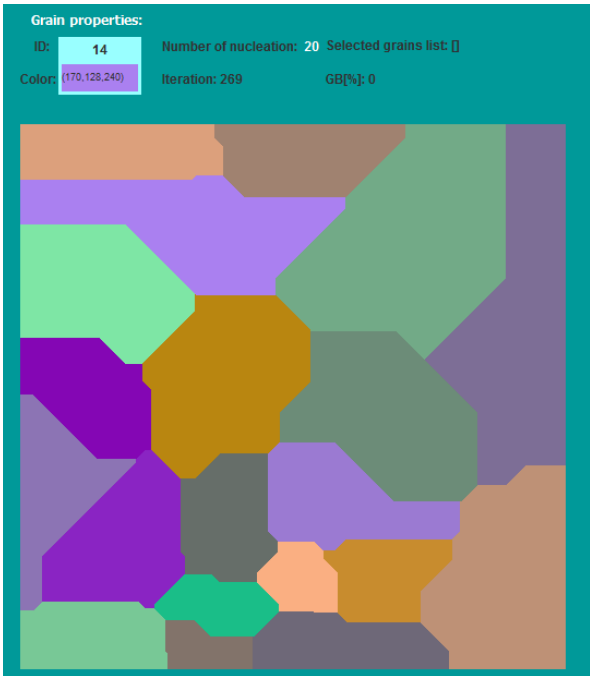

Figure 13. Previewed of grain properties

3. Comparison with real microstructure

In this point compared real microstructure of steel with microstructure generated by the created application.

Fig. 14. The Microstructure generated by the application

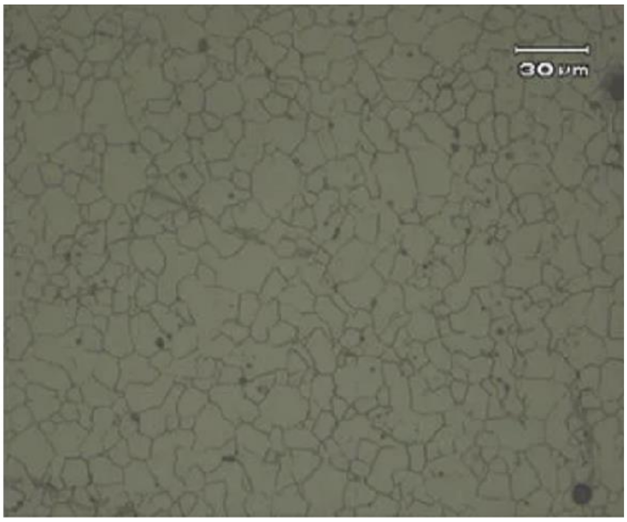

Fig 15. The Microstructure for the as-received AISI 8620 steel Source: [1]

As you can see, both microstructure images are similar. The grain shape of the microstructure created by the program is close to real, as is the case with the grain size, which is similar in both images. The number and size of intrusions is also about the same in the generated as well as the real microstructure.

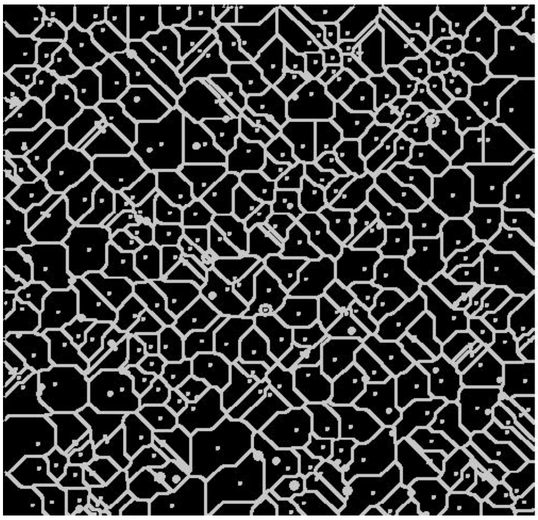

Fig. 16. The Microstructure generated by the application - boundaries and inclusions

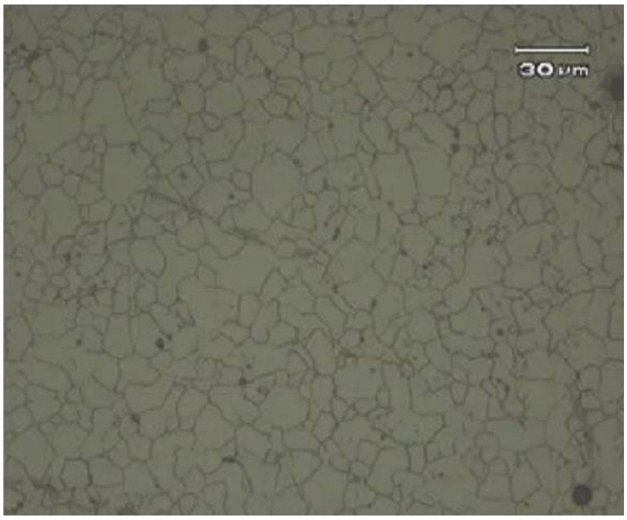

Fig 15. The Microstructure for the as-received AISI 8620 steel only grains Source: [1]

The above drawings show the same microstructures, but in the case of the first structure generated by the program, presented only grains boundaries with intrusions. Although the size of the borders has been set to 1 pixel, they are much thicker than in the real structure.

4. Conclusions

The project uses Cellural Automation algorithm that gives the opportunity to create various microstructures. A cellular automaton is a collection of "colored" cells on a grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many time steps as desired.

In addition to the grid on which a cellular automaton lives and the colors its cells may assume, the neighborhood over which cells affect one another must also be specified. The simplest choice is "nearest neighbors," in which only cells directly adjacent to a given cell may be affected at each time step. Two common neighborhoods in the case of a two-dimensional cellular automaton on a square grid are the so-called Moore neighborhood (a square neighborhood) and the von Neumann neighborhood (a diamond-shaped neighborhood).

The created application enables simple cell growth, changing boundary conditions, adding inclusions, creating borders, and changing the structure. Although the application uses simple algorithms, the results obtained are satisfactory, because comparing the real microstructure with the generated by the program are more or less similar. The shape of the grains, their size, as well as the resulting intrusion are very similar for both cases of microstructures. The disadvantage is the reproduced thickness of the boundaries between the grains.

5. Bibliography

[1] T.Giordani, T.R.Clarke, C.E.F.Kwietniewski, N.I.Kobasko;" Mechanical and Metallurgical Evaluation of Carburized, Conventionally and Intensively Quenched Steels", https://link.springer.com/article/10.1007/s11665-013-0522-2/figures/5?shared-article-renderer