Multiscale modelling

Report

Paweł Chołost

AGH WIMiIP IT

30.01.2020r.

Github: https://github.com/pcholost/Multiscale-Modelling

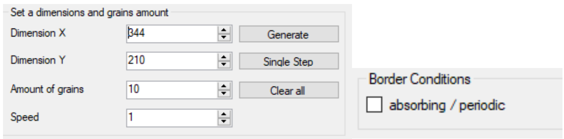
Introduction

Multiscale modelling is a style of modelling in which multiple models at different scales are used simultaneously to describe the system. In this course I use a Cellular Automata technique to divide a specific part of the material into one-, two- or three-dimensional lattices of finite cells. I have learned that this methodology can be used to generate a grain growth. Throughout the course, I had to implement a grain growth with the Moore neighborhood and other improvements which will be presented in the following paragraphs.

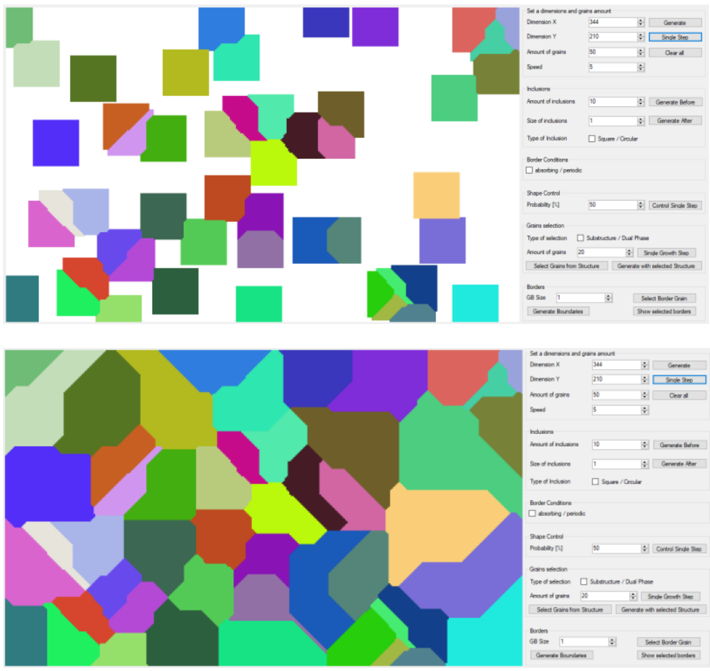
To implement this program I created a Windows Form Application with the C# programming language.

Project 1 – Simple grain growth and import/export file.

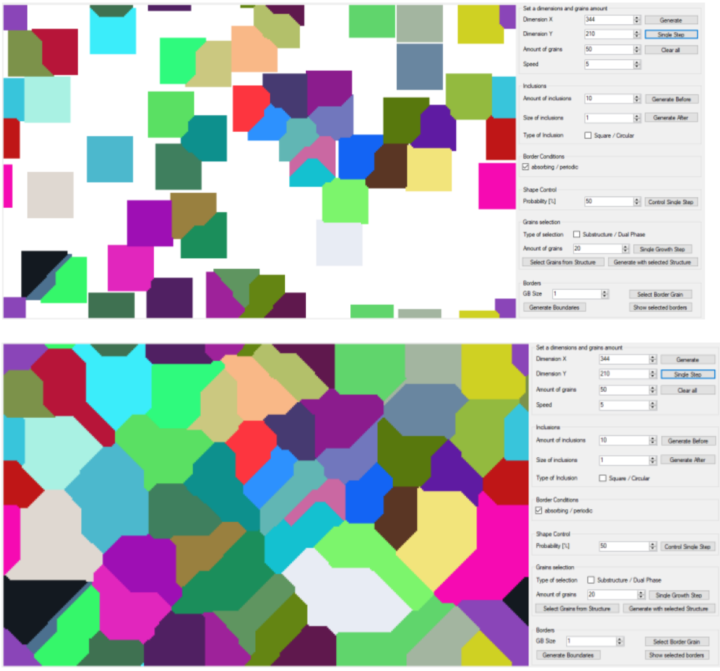
In the first project I created a simple user interface to define the dimensions of bitmap and number of grains which will be generated randomly. User has a possibility to create a new grain by clicking the picture box with generated mesh. The dimensions and all the inputs have a defined default values and limits to reduce out-of-range problems. The function with grain growth take place in the Single step button. User can decide in checkbox which boundary conditions will be used for grain growth.



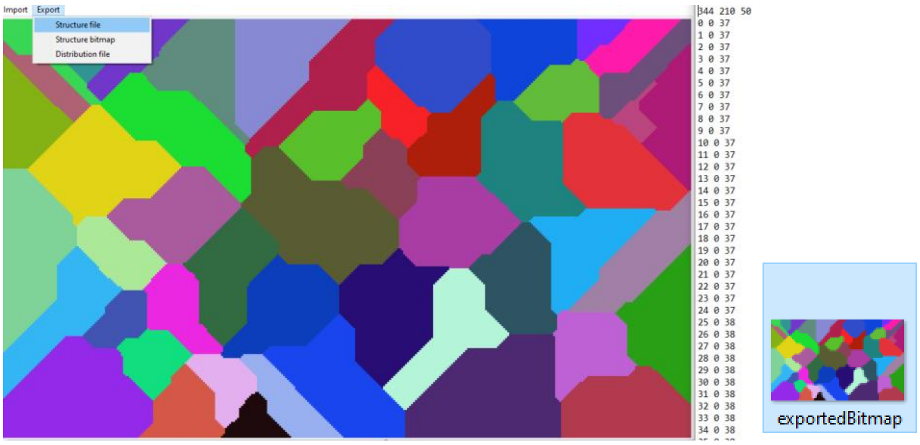
Picture 1. GUI with setting dimensions and amount of grains. Additionally, it is possible to choose the boundary conditions.



Picture 2. Grain growth with absorbing boundary conditions.



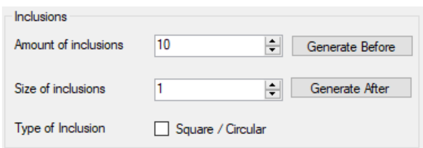
Picture 3. Grain growth with periodic boundary condition.



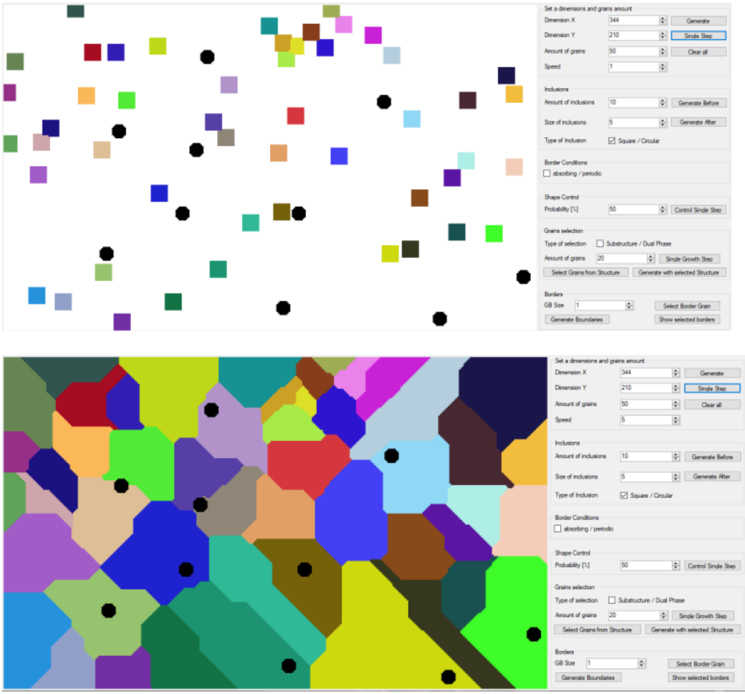
Picture 4. Export generated microstructure to file. Txt file is showing grain coordinates and the value defining the grain. These files can be imported to the app.

Project 2 – Inclusions.

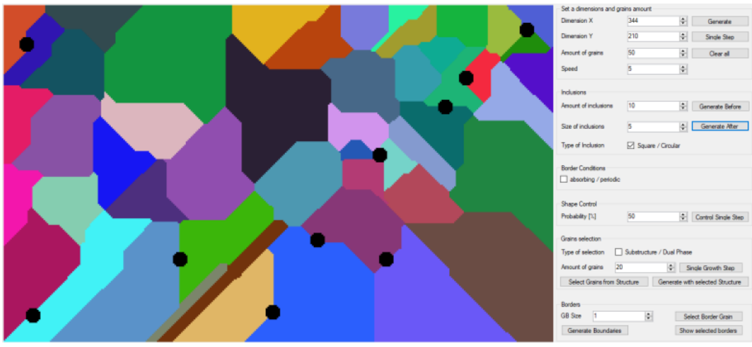
The second project focused on adding inclusions to app. There are two different types and options of generating inclusions that the user can apply. The first option is “Generate Before” - places the inclusions before the grain growth and the grain will not be able to grow in on the placed spots. Whereas “Generate After” sets the inclusions after the completed growth. The condition is that the inclusions must be generated at the grain boundaries. The user has an option to select the type of inclusions from square to circular.



Picture 5. GUI for creating the inclusions.

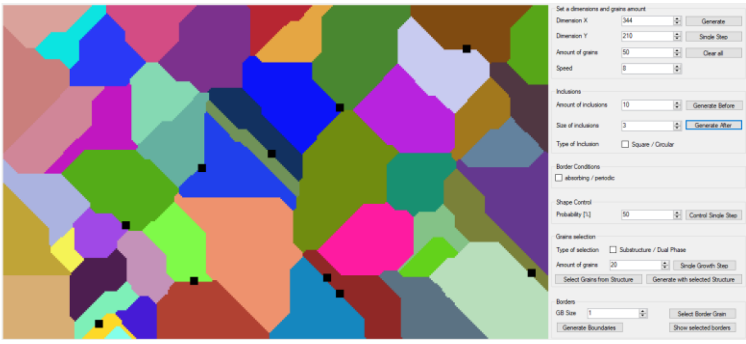


Picture 6. Circular inclusions before grain growing. The size of the inclusions is 5 cells.



Picture 7. Circular inclusions after grain growing. The inclusions are at the grain boundaries.

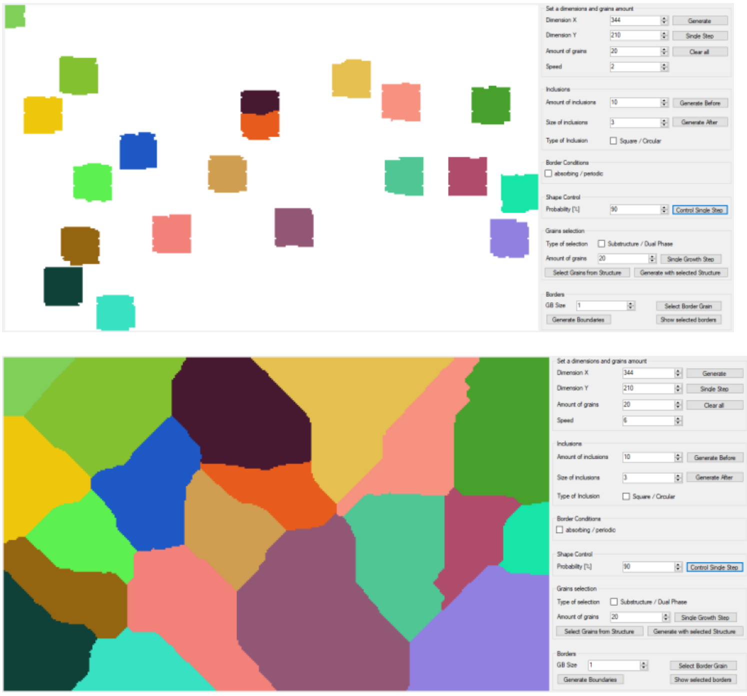
Picture 8. Square inclusions before grain growing. The size of the inclusions is 3 cells.



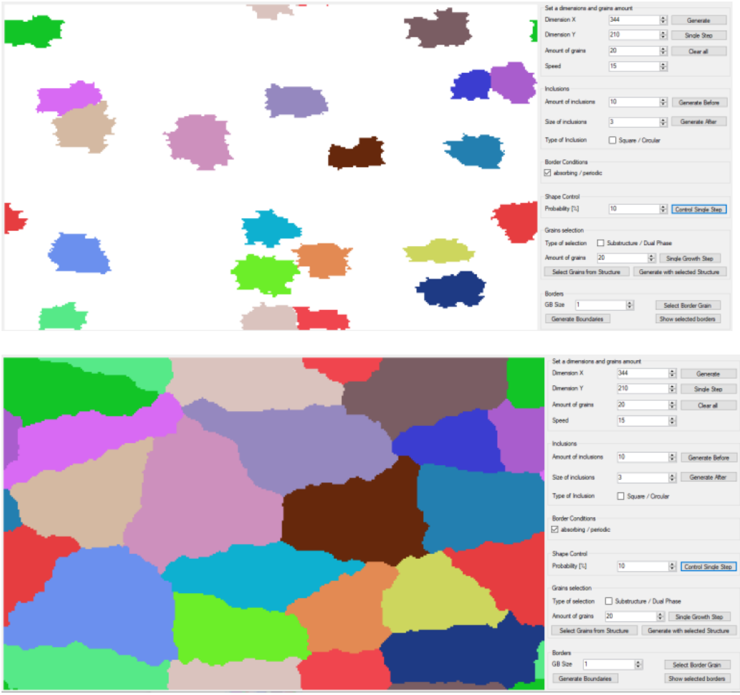
Picture 9. Square inclusions after grain growing.

Project 3 – Grain boundary shape control.

In this section I implemented a new grain boundary shape control for Cellular Automata growth. This grain characterized by the fact that user can personalized the growth by changing the probability value. The algorithm consists of 4 functions combined into one mechanism. The first function called Moore changes the cell value if it has 5 to 8 neighbors of the same value. Next 2 and 3 function will execute when Moore is not fulfilled. It will execute when the call has three or four neighbors in the correct position. If none of the functions will be performed for a specific cell, then the probability function will be executed. This function checks the neighbor with the appropriate probability and changes cell state if the probability agrees.



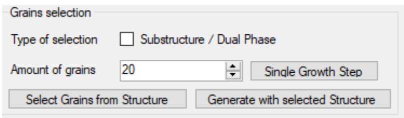
Picture 10. Grain growth with shape control for 90% probability. A absorbing borders condition is selected.



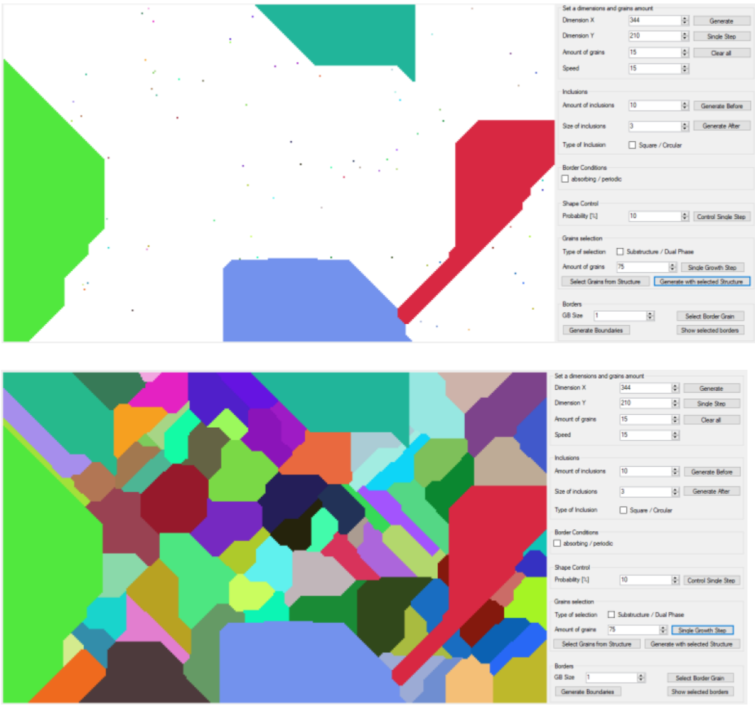
Picture 11. Grain growth with shape control for 10% probability. A periodic borders condition is selected.

Project 4 – Substructure, Dual phase CA.

Project 4 concerned the implementation of second grain growth for two type of selection called substructure or dual phase. After the correct growth of the grains, the user selects the grains from the mesh, which will be kept for the next phase of growing. All other grains that are not selected disappear and are replaced by new by new grains. Grain growth is followed, but without selected grains in the first phase. The main difference between the substructure and dual phase is that in substructure the grain retains its color value, while in the dual phase the grains take the color value of first selected grain. After successful second grain growth, the user can generate the file with percentage grain distribution compared to others.



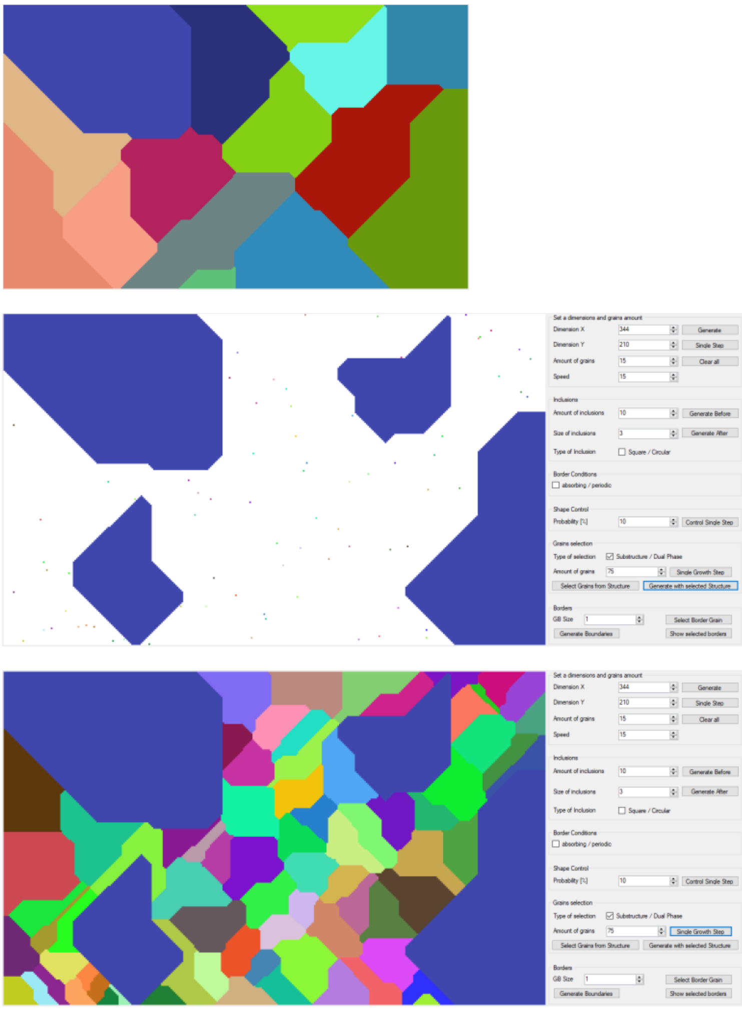
Picture 12. GUI for grains selection to second grain growth.



Picture 13. Second grain growth for substructure. These 4 grains from the first phase remained for the second phase.

```
Id: Size: %
1 4435 6,1393
7 7522 10,4125
10 5185 7,1775
14 3162 4,3771
16 616 0,8527
17 676 0,9358
18 545 0,7544
19 1128 1,5615
20 381 0,5274
21 261 0,3613
22 981 1,358
23 874 1,2099
24 459 0,6354
25 2082 2,8821
26 572 0,7918
27 800 1,1074
28 60 0,0831
29 1674 2,3173
30 754 1,0437
31 552 0,7641
32 158 0,2187
33 488 0,6755
34 358 0,4956
35 110 0,1523
36 507 0,7018
37 1001 1,3857
38 499 0,6908
39 1550 2,1456
40 631 0,8735
```

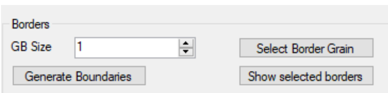
Picture 14. File with percentage distribution. The first 4 values concern the first phase of growth.



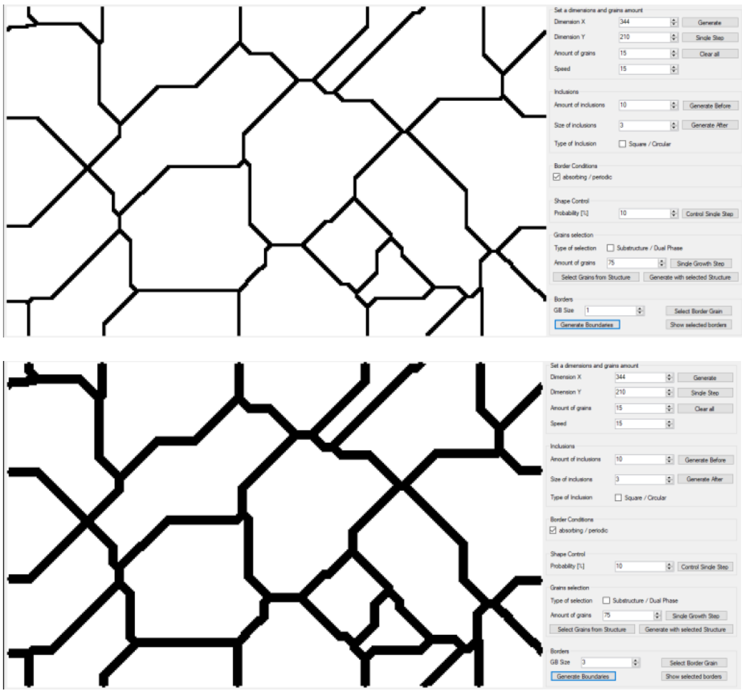
Picture 15. An example of a dual phase selection.

Project 5 – Grain boundaries.

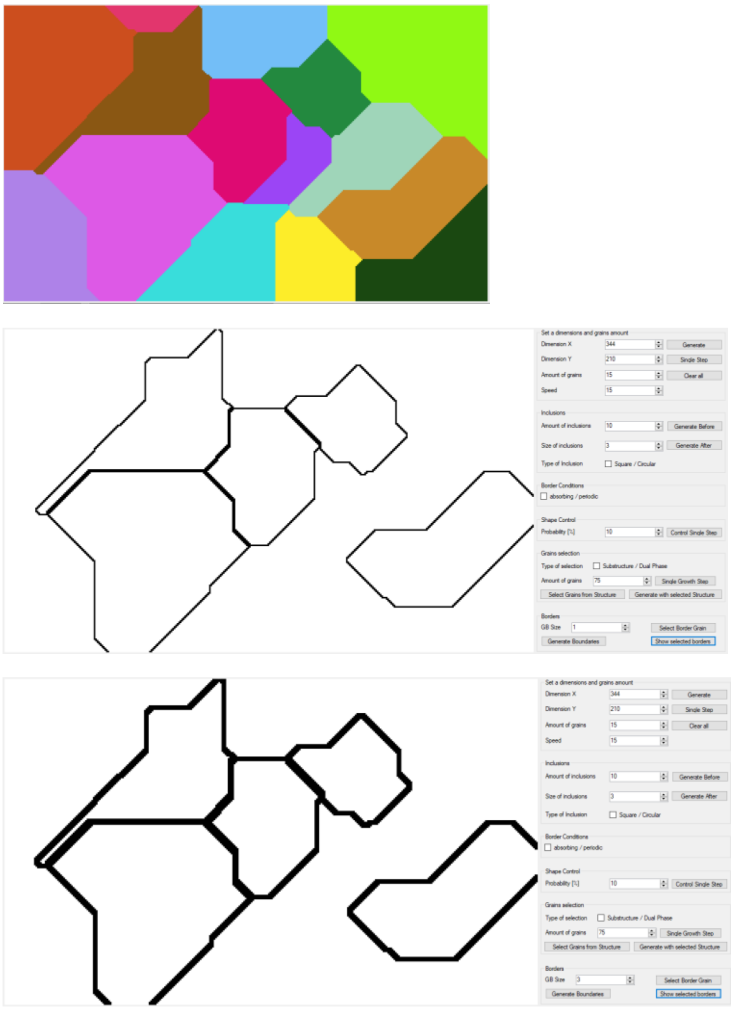
The last part of the project focused on the implementation of grain boundaries. The user can generate the boundaries for all the grains or for selected grains. On user interface it is option to choose a size of boundaries in “GB size” input. The user can easily switch between the CA growth and the generated view of boundaries.



Picture 16. GUI for grain boundaries.



Picture 17. Generated borders for 1GB and 3GB size. A periodic borders condition is selected.



Picture 18. 5 grains were selected and their boundaries was generated. The border size is 1 and 3GB.

Summary.

In conclusion, in this project I have successfully implemented the CA grain growth using the Cellular Automata method with additional functionalities such as inclusions, CA method with shape control algorithm, substructure and Dual phase CA with grain selection and finally displaying the grain boundaries. Implement these functions helped me understand the application of CA method and how it may be applied in practice. The program has the opportunity to develop and implement additional functionalities, for instance, additional neighborhoods, which will allow to simulate various grain growth scenarios.