Multiscale modelling

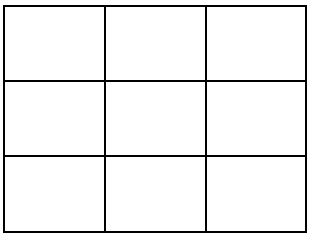
Cellular Automata project report

Author: Gerard Żmuda, IS 2 NS, 2019

1. Introduction

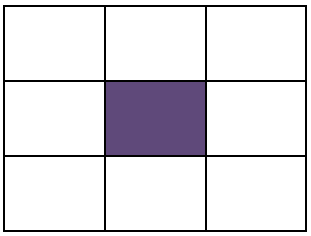
Cellular Automata is one of mesoscale analysis techniques. It was originally developed by Janos von Neuman in 1960s. Created to simulate behavior of discrete and complex systems.

The main concept of this analysis technique is to divide a specific section of the material into one-, two- or three-dimensional lattices of fine cells. CA space is created, containing finite set of cells. Each cell in the space is called cellular automation and is described by a set of internal variables, from which cell state (referenced as “grain id” in implementation) is worth mentioning.



Drawing 1. Sample Cellular Automata space.

During space division, cells are positioned next to each other. Because of that each cell in CA space has its neighbors. Number of adjacent cells can differ based on shape used for space division and neighborhood type assumed.



Drawing 2.Square CA space with Moore neighborhood.

The algorithm of grain growth is based on cell interactions.

First, discrete space is created, composed of a number of cellular automations. All cell states are initialized as void.

In each iteration, cell state of each cellular automation in space is determined by the previous states of its neighbors and the cell itself. The set of generic principles can be established:

- If all neighbors are in void state, the cell remains void.
- If state of required number of neighbors is X, cell state changes to X. Required number might be different based on neighborhood type.
- If there are multiple states (other than void) in neighborhood, cell changes state to one that has the highest presence in the neighborhood.

It is worth noting that above principles are highly generic and should not be treated as grain shape control rules, but rather as generic guideline for algorithm implementation.

The algorithm is finished once there is no void state in whole CA space.

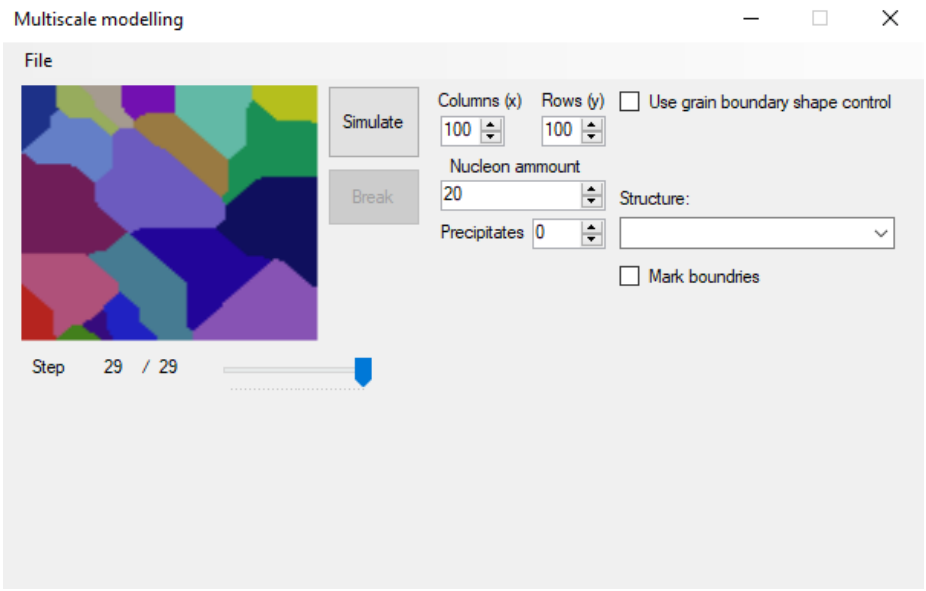
2. Implementation

Fully implemented project can be found at https://github.com/GerardZm/multiscale_modelling

Project has been implemented using C# language.

3. GUI description

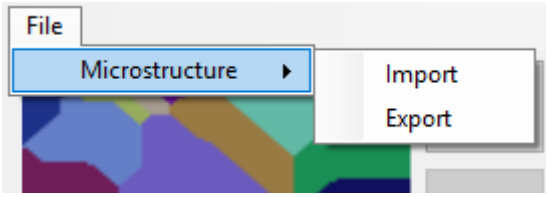
Application uses only a single window to show all components of the project.



Drawing 3. Application window.

Within the window, sections responsible for different tasks are separated.

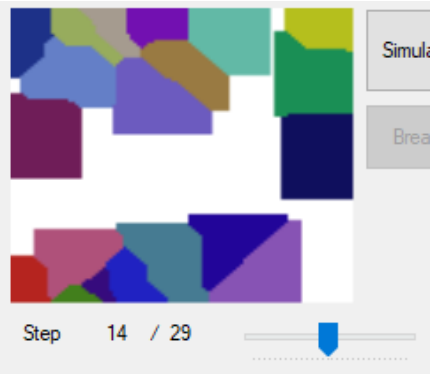
In far left corner, „File" menu is shown, containing importing and exporting activities.



Drawing 4. File menu strip.

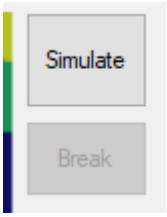
It is possible to import and export results in CSV file format.

On the left visualization tile can be found. This is where simulation progress and results are shown. User can verify each step of algorithm using slider below the image.



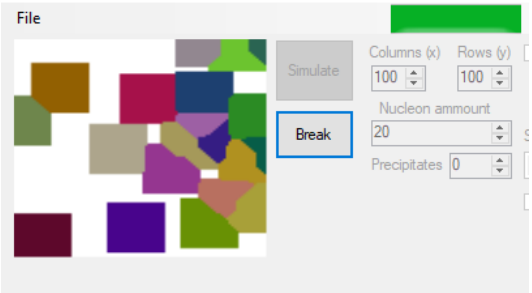
Drawing 5. Visualization tile with slider allows User to check each step of grain growth algorithm.

Moving to the right, Simulate and Break buttons are shown. These are used respectively to start the simulation and stop it on User's demand.



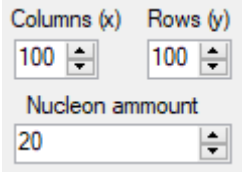
Drawing 6. Simulate and Break buttons.

Break button is enabled only while calculations are in progress.



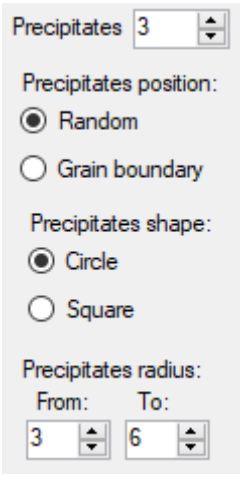
Drawing 7. Break button enabled during calculations.

On the right of buttons there are three fields, Columns(x), Rows(y) and Nucleon ammount. This is where User defines space size and number of nucleons (grain ids) to be assumed in calculations. An example is shown on drawing 8, using space of 100x100 cells and 20 nucleons.



Drawing 8. Selection of CA space size and number of nucleis.

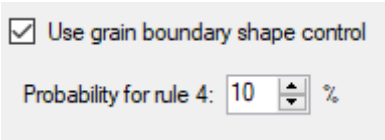
Below selection fields, precipitates section is shown. Once User selects at least one precipitate to be assumend during calculations, an additional sections shows up, allowing to customize the global parameters of precipitates, as shown on Drawing 9.



Drawing 9. Precipitates customization section

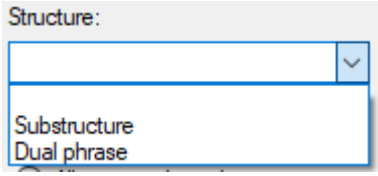
If User selects random position, precipitates are added to the space during the initial step of algorithm. Otherwise, if grain boundary position is selected, precipitates are added in the end of calculations.

On the right, User can select an option to use grain boundary shape control. The shape control algorithm consists of 4 rules and latter of them requires probability level for calculations. User can define the probability by using field that appears below the checkbox, as shown on Drawing 10.



Drawing 10. Grain boundary shape control with probability selection.

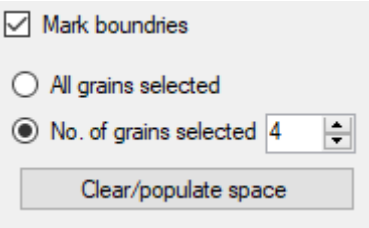
Below shape control section, structure selection section can be found. Substructure and dual phrase structures can be acquired using this dropdown selection.



Drawing 11. Structure selection section.

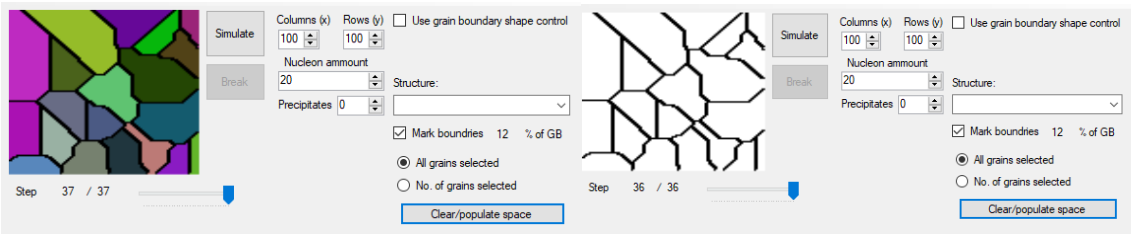
If User does not intend to use one of defined structures, this option has to be left empty.

The last section, positioned below structure selection section is the grain boundary marking section. This section allows User to mark the grain boundaries in the end of calculations. Either all grain boundaries can be marked, or only a selected ammount of them, based on selection made by User.



Drawing 12. Boundaries marking section.

Using button „Clear/populate space" shown in Drawing 12, User can select to show or hide the colorful structure. The example of showing and hiding actions are shown in Drawing 13.



Drawing 13. „Clear/populate space" button allowing to hide or show the structure.

4. Results

Multiple microstructures imitating the real representatives can be acquired using the program.

Examples:

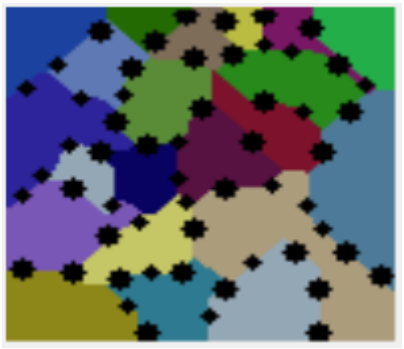
- Simple microstructure without grain boundary control and precipitates

- Microstructure with precipitates, without grain boundary control

- Microstructure with grain boundary control (probability 10%) without precipitates

- Microstructure with precipitates and grain boundary control (probability 20%)

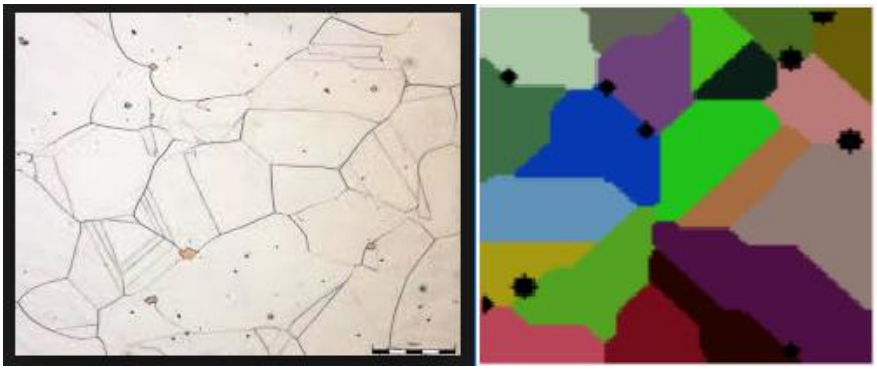
- Dual phrase microstructure with precipitates



5. Comparison with real microstructures

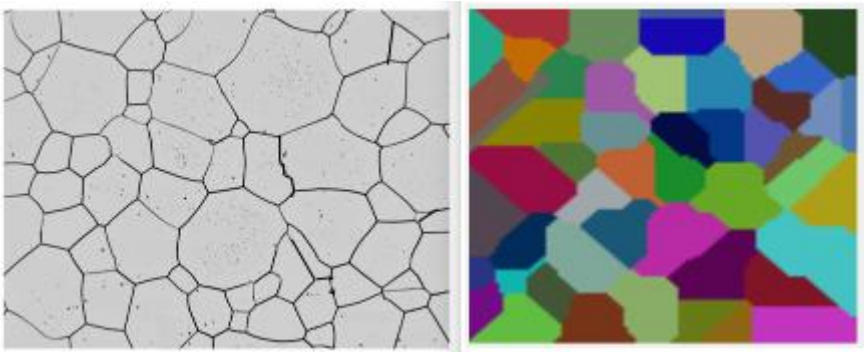
This chapter contains comparison of microstructure acquired from application and real representitives.

- Austenitic steel X8CrNiTi18-10



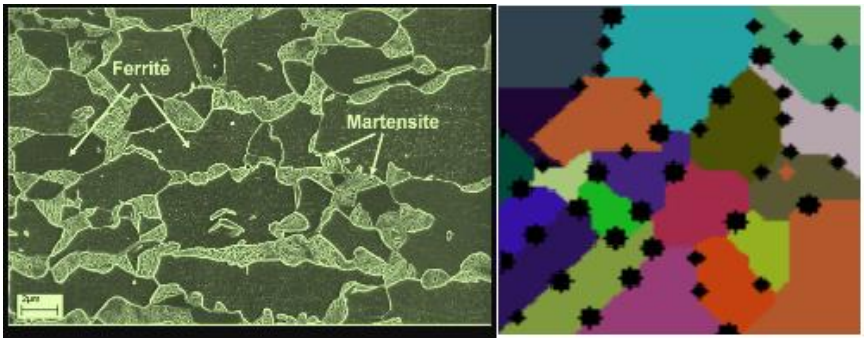
Grain shape is really close between real example and simulation. Occasional precipitates are located at grain boundaries on both drawings.

- Annealed low carbon steel



Grain shape looks simillar, but simulation shows lack of really small grains compared to the real example.

- Dual phrase steel



Larger and smaller grains are present in both cases. Martensite precipitates are similar (although they are elongated in case of real structure, which was not done in case of simulation)

6. Summary and conclusions

During Multiscale Modelling practical classes I have learned how to implement Cellular Automata algorithm, showing real microstructure creation process modelled by digital calculations.

The application shown in this report is able to effectively imitate the real microstructures by using Cellular Automata method. The implemented parts of computational algorithm have been provided to User.

By further development of the application, it would be possible to imitate microstructures acquired during multiple metal forming processes. The project could be also adjusted for simulation of synthetic materials, ceramics and composites. All of this would allow to create microstructure that could be verified and approved by meterial engineers without industrial try-out process. This would lead to reduction of production costs and time required to put new material into production.