Multiscale Modelling

Second part report

Author: Wojciech Mikuś
Date: 16.01.2019

1. Short introduction

Program has been made as an extension of the previous, first part described in report 1. It is developed in the same technology, using the same set of tools. This report contains description covering only material from 8^{th} to 12^{th} classess.

2. Used technology

Author decided to implement said application using **Java** programming language. Although there are quite a few possibly better choices (in terms of efficiency), Java offers an unique easiness of launching application on different systems supporting Java without any additional tools required. Also, personally author believes Java to be more friendly for beginner-to-intermediate level of developers than more advanced, lower level languages like C.

Every step of developing this project has been made using Windows 10 operating system. Concerning graphical user interface, author has chosen **JavaFX** library, simply because it is newer than highly popular Swing. Also, JavaFX won the heart of the author by providing separate XML files for design of GUI, and those files look relatively close to XML files used in web development. Scene builder has been used to create GUI.

As a version control system author has decided to use well known all over the world github, because of it reliability and user friendliness.

Code is avaiable on https://github.com/sophosus/Multiscale-Modelling

3. Program and its user interface

As previously, Application consists of single window in which user can specify conditions of simulation. Also, in the same window there's a special reserved place for printing result. More detailed descriptions will be presented in following subsections.

- **Main window**

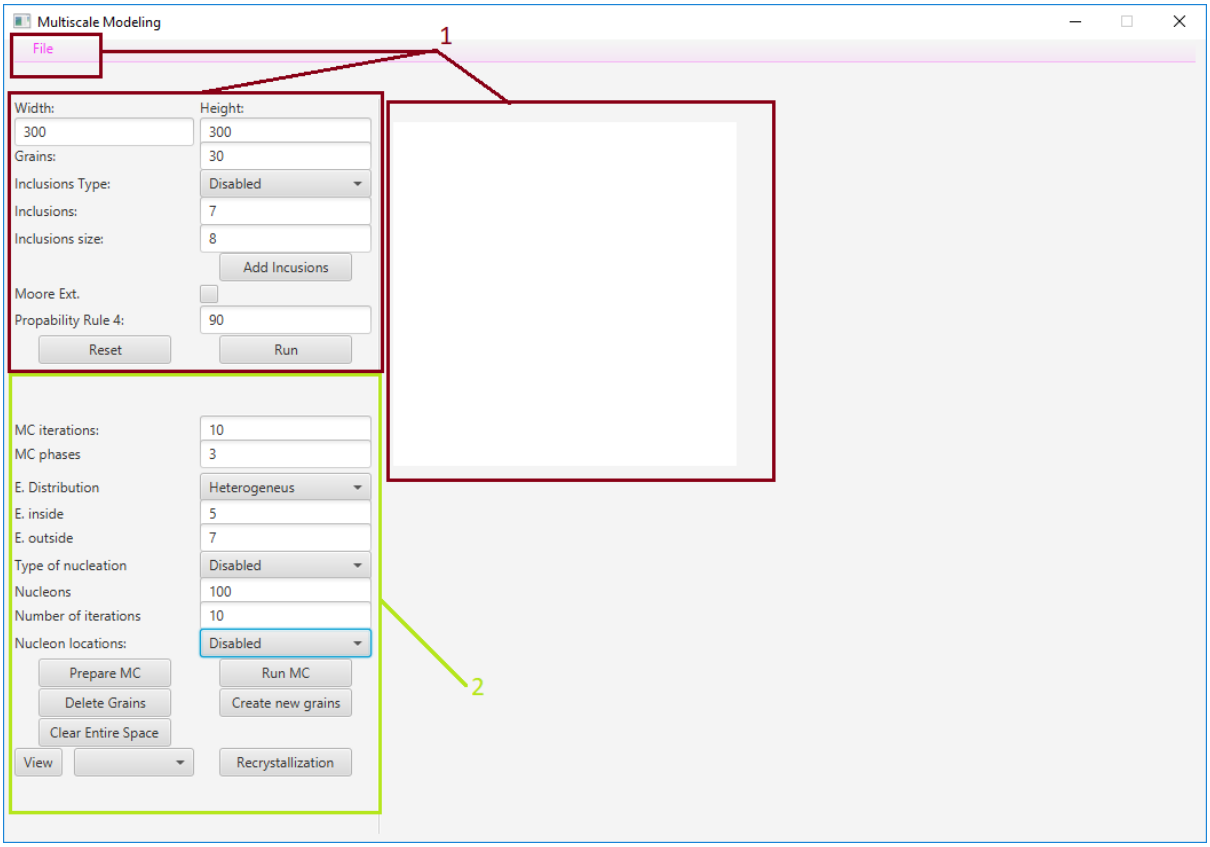

Image 1. Graphic user interface

Options contained in red square have not been changed since first report, thats why author decided to not describe them again.

New functionalities added to program have been marked with light green square. Description with close-up can be found below.

- **Simulation parameters:**

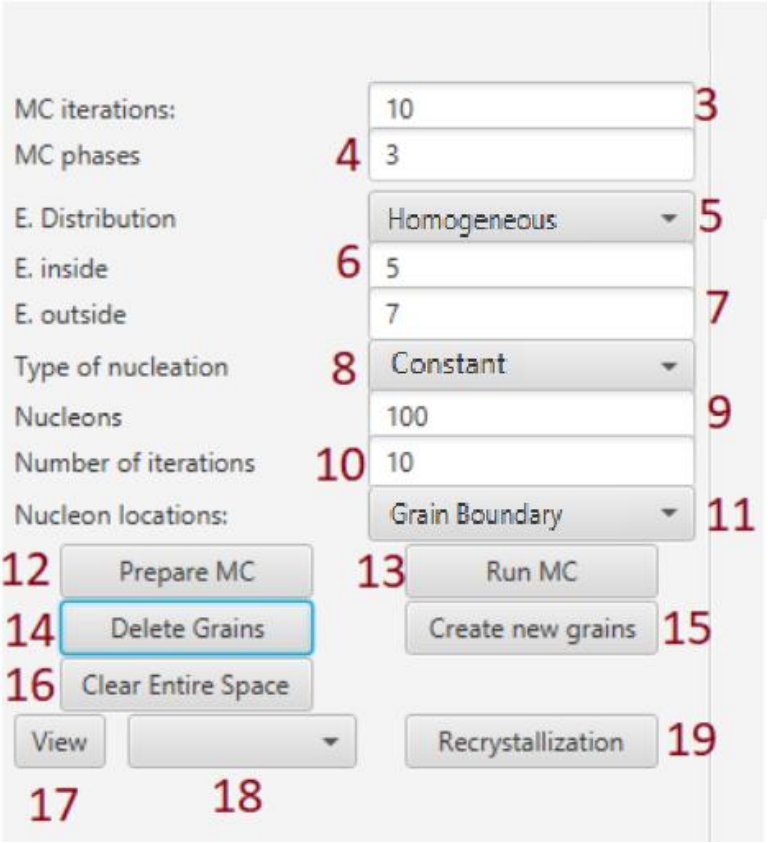

Image 2. Simulation options- GUI

Beginning from the top:

3. Number of Monte Carlo iterations
4. Number of Monte Carlo phases
5. Energy Distribution choice list : Homogenous or Heterogeneous
6. Energy inside grain (lower)
7. Energy on grain boundary (higher)
8. Possible types of nucleation : at the beginning, constant and increasing
9. Number of nucleons added in recrystallization process
10. Number of iterations of recrystallization process
11. Location of nucleons in the recrystallization process
12. Button for initialization of Monte Carlo method
13. Button used to launch Monte Carlo method with a chosen amount of steps.

14. Button which deletes all but selected grains (grains should be selected by mouse pressing on their location)
15. Button which generates new grains, use it after button (14)
16. Button which clears everything but grain boundaries.
17. Button to change between views chosen on list (18)
18. list of possible views: Default, Energy and recrystallization.
19. Button for starting recrystallization process.

- **Example simulations**

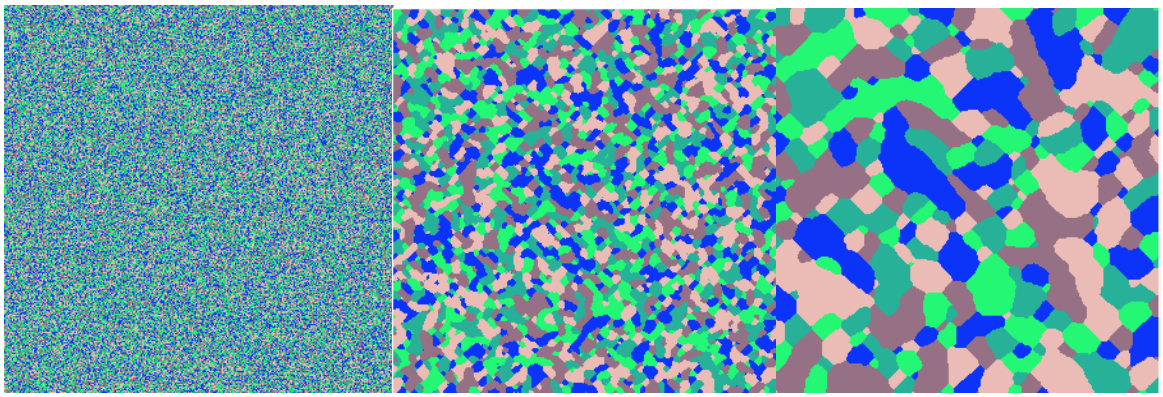

Image 3. From the left: 1. Initial space with 5 possible phases. 2. After 10MC steps. 3. After 100 MC steps

Images shown above depict process of simulation using Monte Carlo method. As can be spotted, the more MC steps, the less individual grains remain in the material.

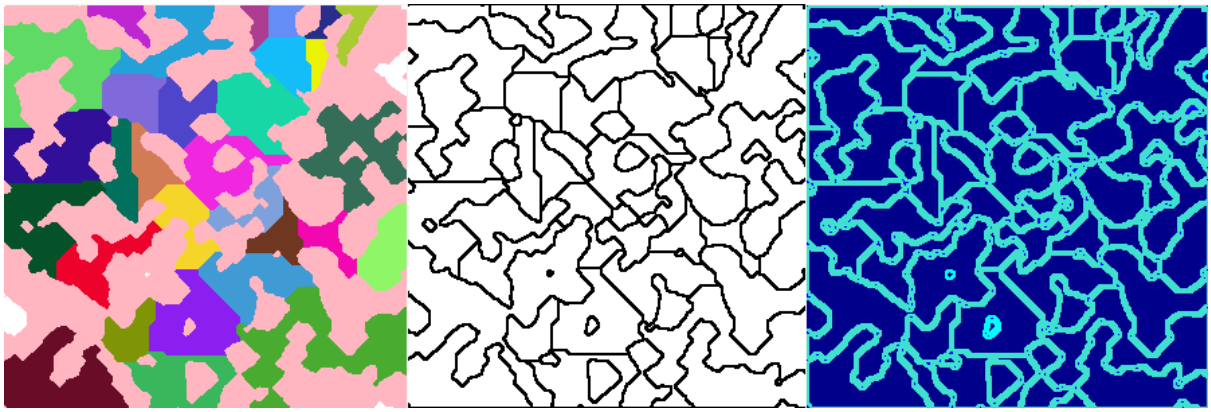

Image 4. From the left: MC->CA, Clear entire space, Energy view for final result

Image 3 represents respectively from the left: 1. results for few grains expanded using MC. Second phase has been made from those grains, and then followed CA simulation for 30 new grains.
2. Functionality of 'Clear entire space' button. 3. Energy view for generated structure with heterogeneous energy distribution. Energy values equal to 5 inside and 7 on the grain boundaries.

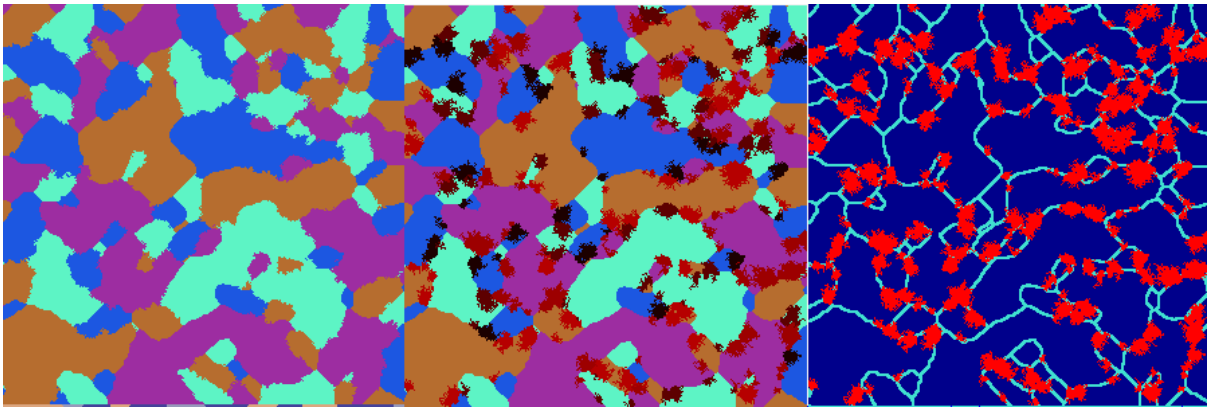

Image 5. 150 steps of MC followed by 3 steps of recrystallization – possible views

Image represents possible views generated by application. Every picture represents the same structure- 4phases after 150 MC steps and addinational 3 recrystallization steps with 120 new nucleons located on grain boundaries. On The first one from the left the view mode is set to 'default', middle image represents the 'recrystallization' view mode, and the one on the right depicts 'energy' view mode. Turquoise color stands for higher energy value, dark blue corresponds a lower energy.

4. Comparison with real microstructures

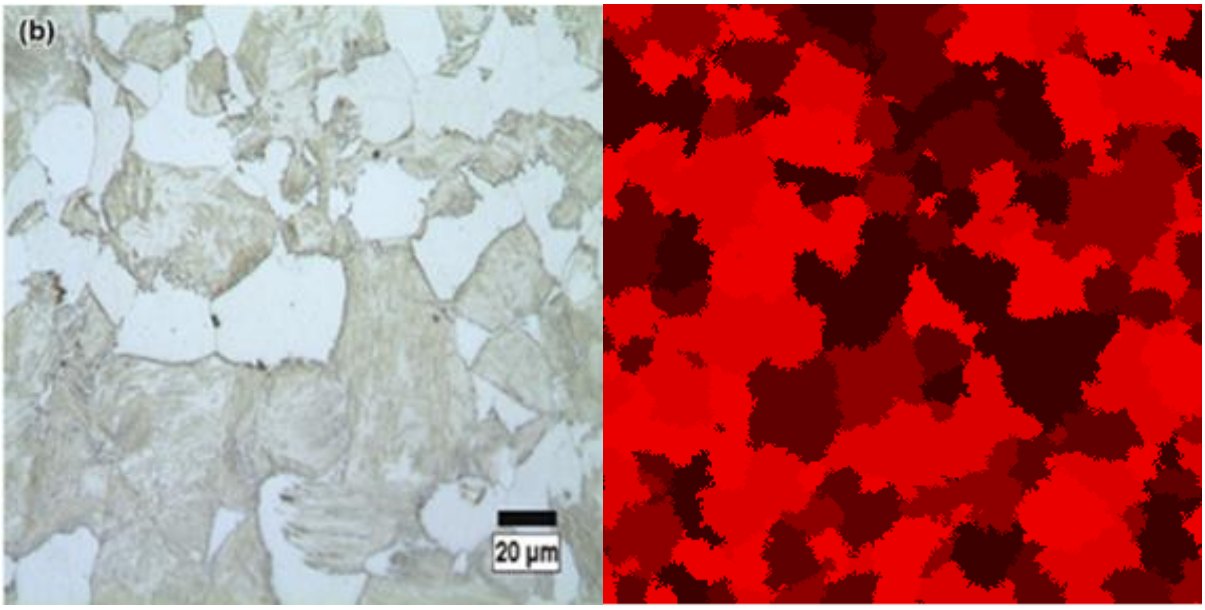

Image 6. Recrystallized dual phase steel comparison, original and program made.
Source: https://www.researchgate.net/figure/Microstructures-of-dual-phase-carbon-steel-intercritically-annealed-at-775-C-a-fibrous_fig3_272966423

Despite visible jagged edges, simulation's result shows some level of resemblance to dual phase steel microstructure

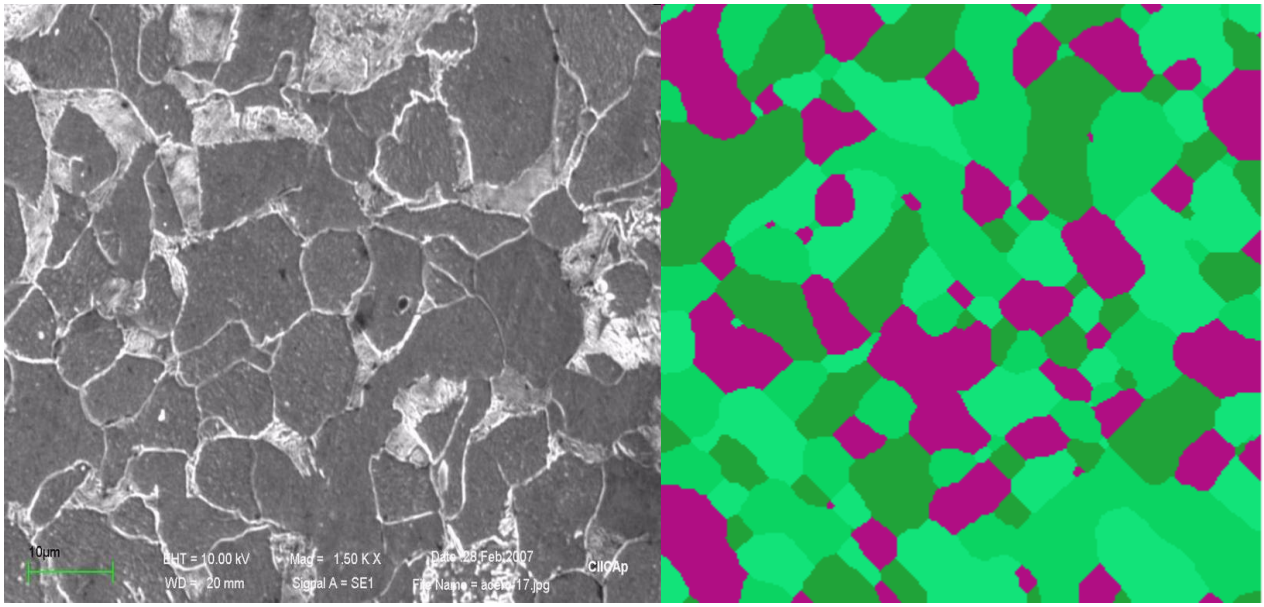

Image 7. Recrystallized dual phase steel comparison, original and program made.
Source: https://www.researchgate.net/figure/Optical-images-of-the-microstructure-of-carbon-steel_fig1_266258854

Second example shows striking similarity to real dual phase steel. To achieve this effect, program had to calculate 150 steps of MC method

5. Summary

As an overall conclusion, application is capable of simulating real, existent structures, but of course not all of them. Many real structures still can be generated using cellular automata, but at least some level of knowledge in crystallography is recommended in order to achieve desired effects. Even having that, it might just not be enough. That is why Monte Carlo method has been implemented. It is an useful extension of the first part of the program, because of greater flexibility of simulation conditions. Addition of energy model is another quite good feature of this part of application. Because of it, there is a possibility to simulate at least some recrystallization processes.