Multiscale Modelling - report

Author:

Daniel Półchłopek

1. Technology

Application was written using Java. Java is a programming language and computing platform released by Sun Microsystem. There are a lot of applications that run with Java and new ones are created every day. Java is mainly used in servers, game consoles, cell phones. One of the biggest advantages of Java is its portability. We can run the application written on Windows on other operating systems such as Linux or MacOS. This is possible by using JRE (Java Runtime Environment).

In my project I also used JavaFX, which is GUI library. Oracle declared that from Java 8, JavaFX becoming the recommended library for creating graphical user interface (earlier it was Swing and AWT).

Scene Builder was also used to create the application. JavaFX Scene Builder is a tool which lets users in fast way design JavaFX application user interface, without coding. In this tool programmer can drag and drop components to a work area. Users can also modify their properties. When they finish, the code is automatically generated to a file with the .fxml extension, which can be linked to the JAVA project by binding the UI to the application's logic.

The integrated development environment I chose was IntelliJ. IntelliJ is written in Java for developing computer software. It is developed by JetBrains. The first version was released in January 2001, and was one of the first available Java IDEs with code refactoring capabilities integrated and advanced code navigation.

2. GUI description

JavaFx allows in simple and fast way create simple GUI contains basic components like: buttons, text fields, choice boxes or place to draw any shapes (in application to draw board was used canvas component).

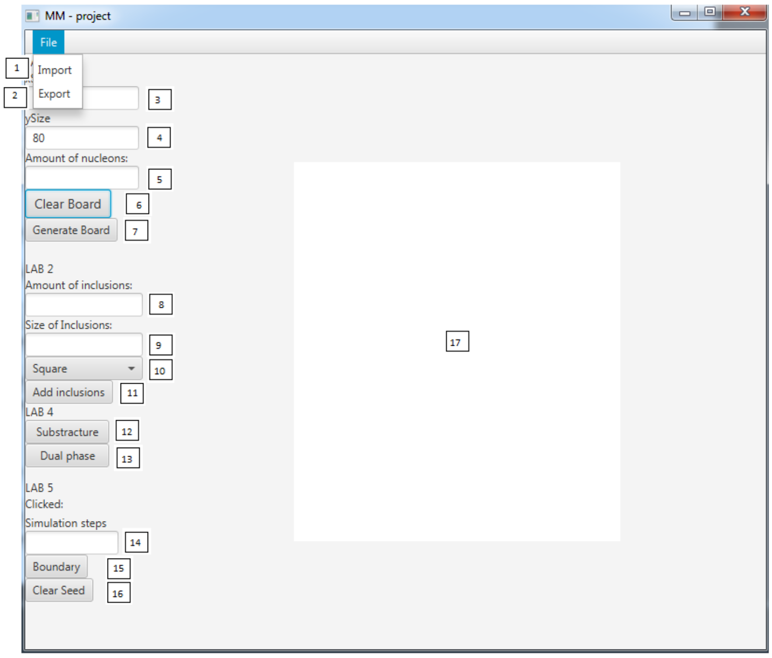

Figure 2.1 GUI

Application GUI is shown in Figure 2.1. We can see 17 different components, that have been implemented to cover demanding from 5 laboratories.

Behaviour of all component is described below:

1. Import – click this option causes that FileChooser component pops up, which allow us to choose file to import. After selecting the file, his content will be loaded to our application and will show up on central position window.
2. Export - click this option causes that FileChooser component pops up, which allow us to choose place on computer disk and file extension. An application allows us to choose extensions: .txt and .bmp. Content of file is loaded from canvas.
3. xSize text field - allows to type horizontal canvas dimension
4. ySize text field - allows to type vertical canvas dimension
5. Amount of nucleons text field - allows to type amount of nucleons, which will be generated in canvas
6. Clear Board button - click this button causes clear canvas
7. Generate Board button - click this button causes random grains position and start calculate grain growth algorithm. To generate canvas 'xSize' text field, 'ySize' text field and 'amount of nucleons' text field must be filled.
8. Amount of inclusions text field - allows to type amount of inclusions to generate on canvas

9. Size of inclusions text field - allows to type size of generated inclusions. When type of inclusions type is chosen to ‘Circle’ field determine inclusions radius. In the other case, describing field determine square side.
10. choice box component - allows to choose shape of generated inclusions. Application allows to choose one of two inclusions type - square or circle.
11. Add inclusions button - click this button causes appearing of inclusions on canvas. When canvas is empty, inclusions location will be randomize. In case when canvas isn’t empty, inclusions location will be set on grains boundaries.
12. Substructure button - click this button causes erase all non-clicked grains. Grains, which was clicked will not be erased and their color will be remain.
13. Dual phase button - click this button causes erase all non-clicked grains. Grains, which was clicked will not be erased and their color will be change to pink.
14. Simulation steps text field - allows to type number of steps which grain growth algorithm calculate
15. Boundary button - click this button causes appear boundary for every grains
16. Clear grains button - click this button causes erasing all grains from canvas. If we earlier run simulation and clicked button “boundary”, grains will be removed but their boundaries will remain.
17. Canvas - main application part, where nucleons, boundaries, inclusions are drawn

3. Results

In application there is implemented grain growth algorithm with parameter probability. For different value of probability we can obtain different grain appearance. Pictures placed below show two cases. In figure 3.1 we can observe grains boundary shape with probability equals 90%, in figure 3.2 we can observe grains boundary shape with probability equals 50%. Based on these figures, we can tell that algorithm works correctly.

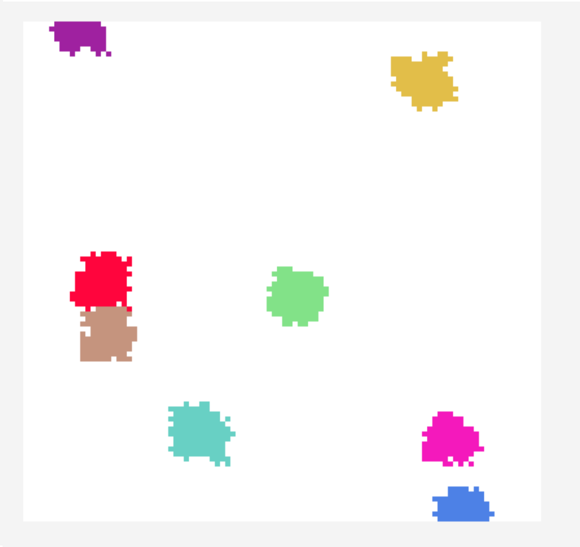

Figure 3.1 Grain growth with probability 90%

Figure 3.2 Grain growth with probability 50%

Next task in project was added possibility to generate inclusions. Generation of inclusions should have different behavior dependence on the moment, when inclusion will be add. For the start of simulation, location should be generate in random way. It is showed in figure 3.3. For the end of simulation, inclusion location should be generate on the grains boundaries. It is showed in figure 3.4.

Figure 3.3 Inclusions generate after simulation

Figure 3.4 Inclusions generate before simulation

Another feature in project is possibility to add boundaries to generated grains. In figure 3.5 we can see boundaries, that have been generated for all grains in simulation. In figure 3.6 we can see boundaries that remain after grains removal. In the figure 3.7 boundaries have been generated for selected grain. In figure 3.8 there are boundaries for selected grain after removal all grains.

Figure 3.5 Generated boundaries without choose any grain

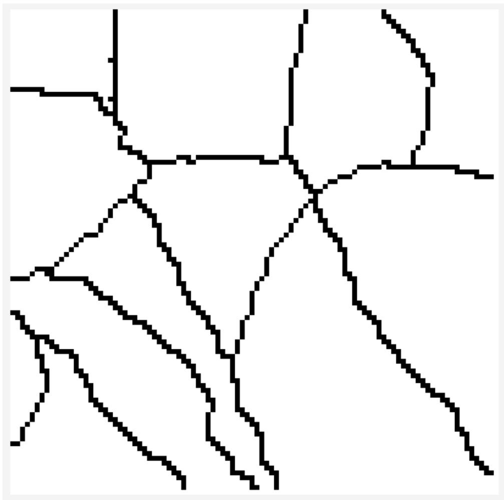

Figure 3.6 Boundaries after grains removal

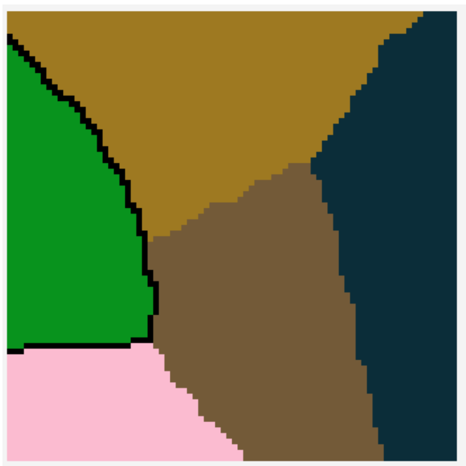

Figure 3.7 Generate boundary with choose one grain

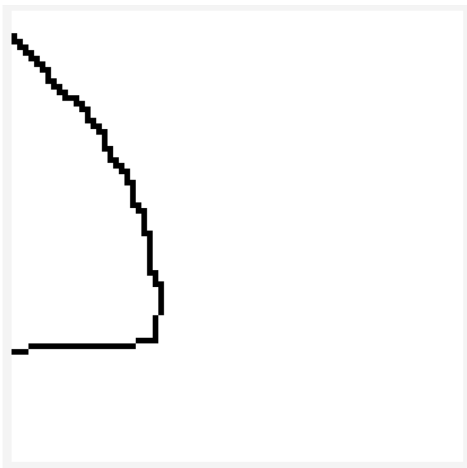

Figure 3.8 Generate boundary with choose one grain, next delete grain

The application has also dual-phase and substructure options to simulate real structures in better way.

First option, we can see on figures 3.9-3.11. First of them(figure 3.9) shows first grains generation. Second(figure 3.10) shows selected grains. Third (figure 3.11) shows second grains generation with selected grains in previous generation.

Figure 3.9 Substructure option, 1st generation

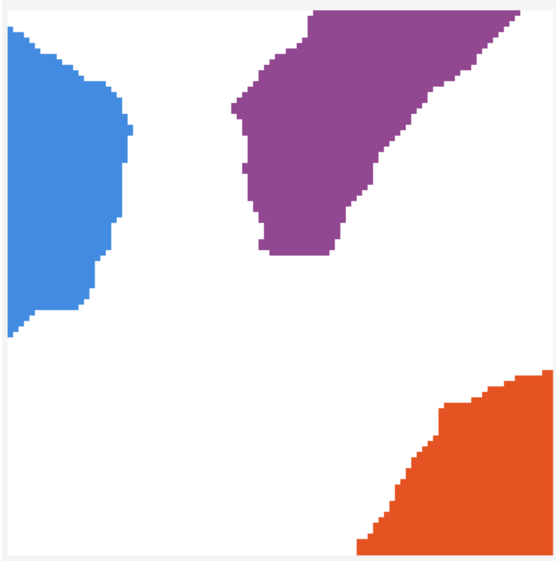

Figure 3.10 Substructure option, grains selection

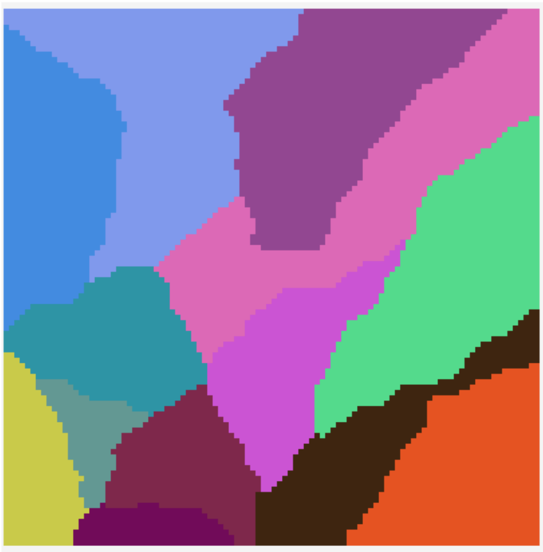

Figure 3.11 Substructure option, 2nd generation

Second option, we can see on figures 3.12-3.13. First of them (figure 3.12) shows first grains generation. Second (figure 3.13) shows selected grains. Third (figure 3.14) shows second grains generation with selected grains in previous generation.

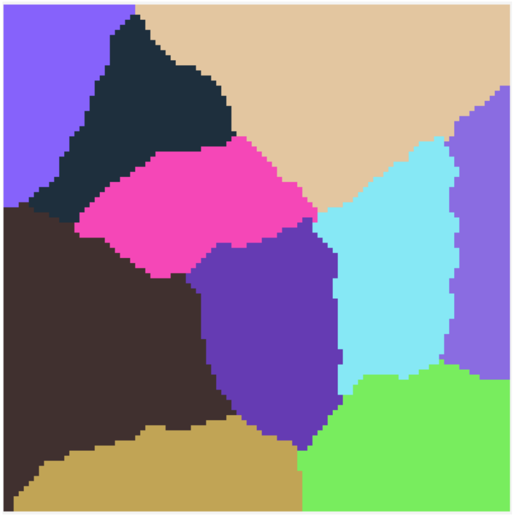

Figure 3.12 Dual-phase option, 1st generation

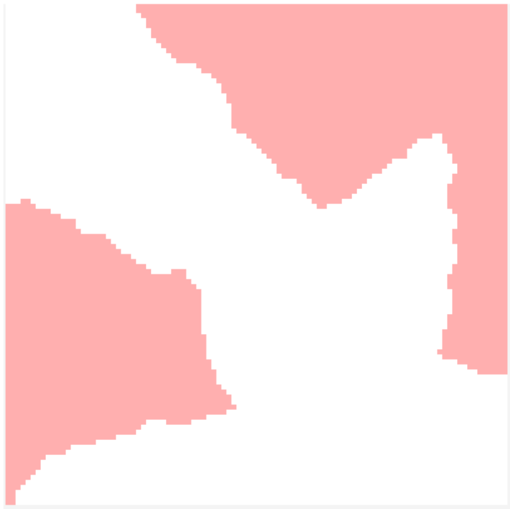

Figure 3.13 Dual-phase option, grains selection

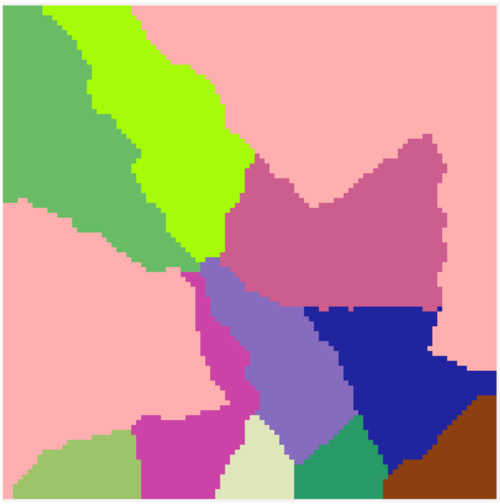

Figure 3.14 Dual-phase option, 2nd generation

4. Comparison with real microstructures

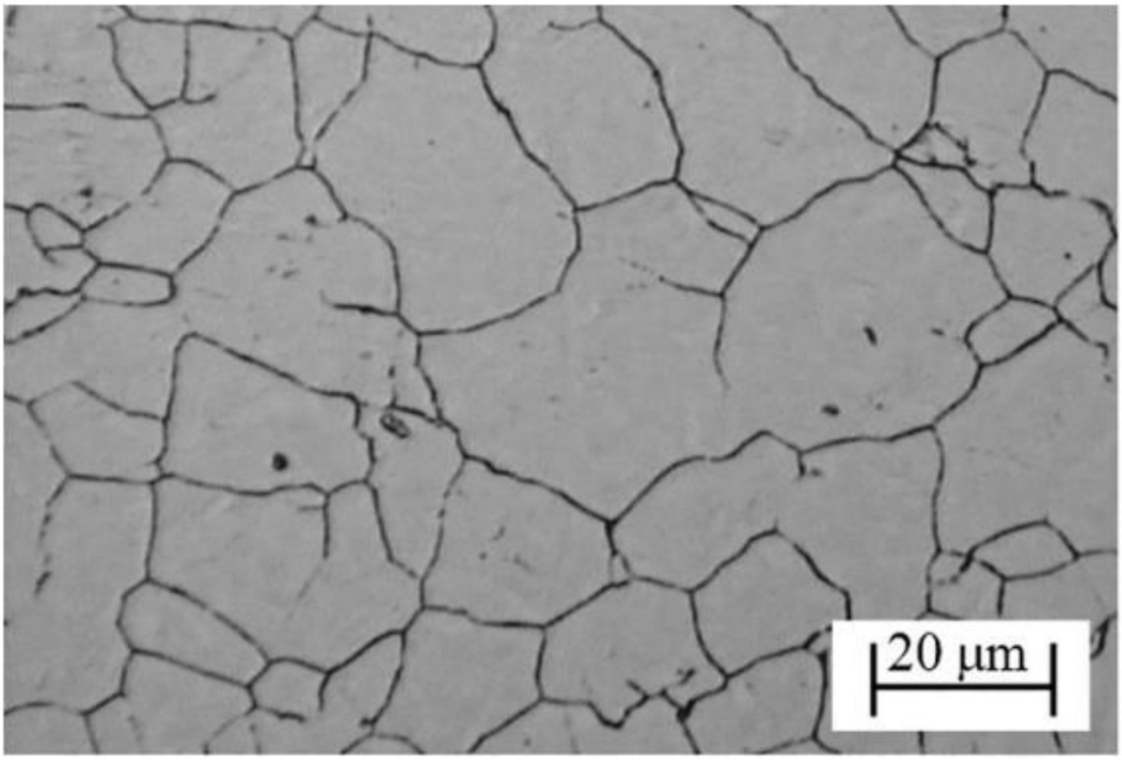

Figure 4.1. Microstructure of specimen of SAE 4340 Q1060

http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-14392017000800039

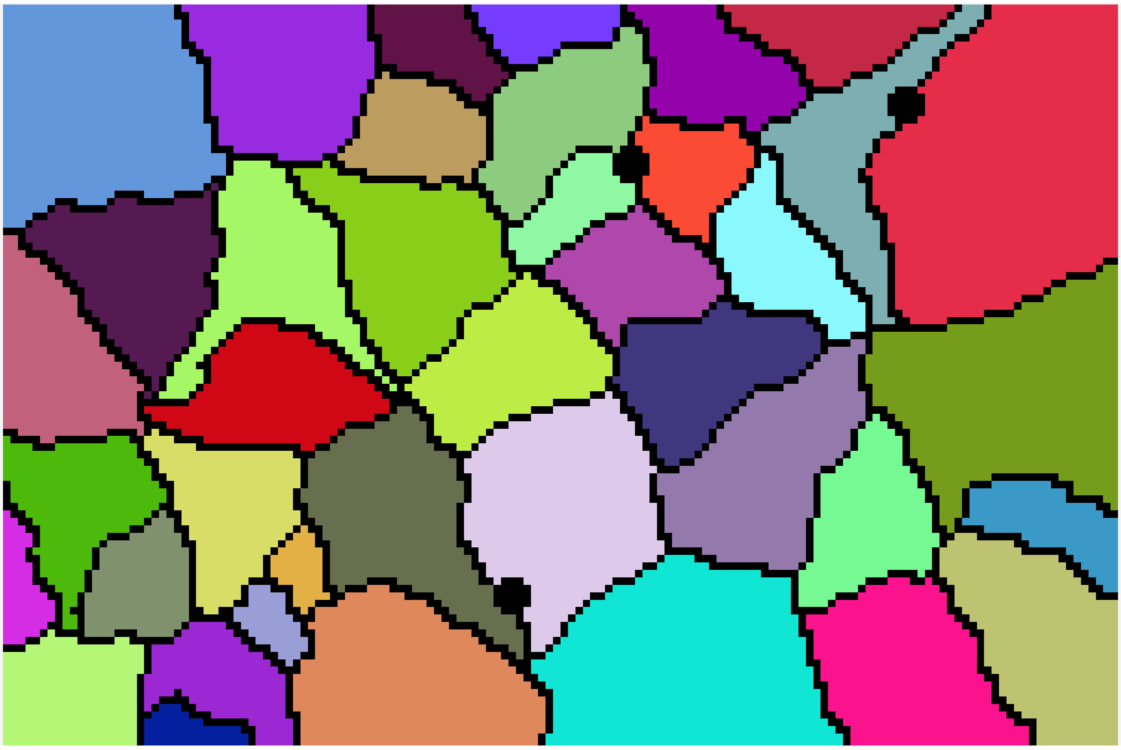

Figure 4.2 Generated microstructure

In this part we compare real microstructure, shown in figure 4.1, with microstructure generated by application, which is shown in figure 4.2.

Based on both figures we can conclude:

- Some of grains generated by application have similar shape compared to those in real microstructure but are located in different place,
- Inclusions in figure 4.2 are much bigger than inclusions in figure 4.1. There aren't located in the same place among other in the figure 4.2 inclusions was added after simulation.
- Boundaries of grains in figure 4.1 are thinner than in figure 4.2. In both figures they are clearly visible.
- One of the reasons for the differences between two figures may be use of inappropriate neighborhood for this case.

5. Conclusions

Implemented application allow the simulation of grain growth using CA. Application contains many useful features for generating microstructures, including: possibility of adding square and circular inclusions of different sizes, determining the number of nucleons, or adding grains boundaries.

Part of generated microstructures in some aspects are very similar to real ones but not ideal. It can be caused by large number of parameters which affect on shape of microstructure (for example neighborhood, amount of inclusions and their shape).

All the project objectives have been achieved. Application is easy to use and thanks to used technologies can be run on other operating systems such as for example Linux.