Multiscale Modelling

First part report

Author: Wojciech Mikuś
Date: 15.01.2019

1. Short introduction

Sole goal of this project was to implement application for grain growth simulation of steel microstructure with use of Cellular Automata methods. Final application allows modification of few simulation parameters, for example dimensions or number of initial grains at the beginning of simulation. Report below contains actual look of application, its graphic user interface and also results of example simulation compared with real microstructures.

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.

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

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 description will be presented in following subsections.

- **Main window**

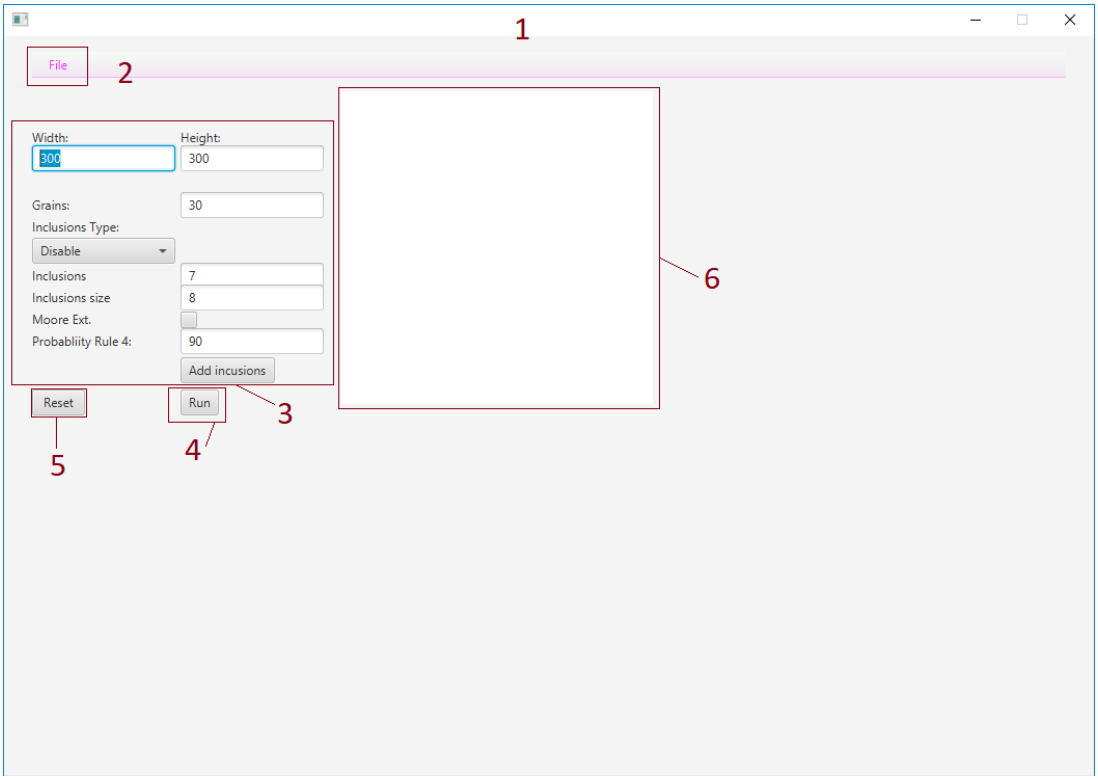

Image 1. Graphic user interface

1. Main window of application
2. ‘File’ menu, allows importing and exporting of.txt and .bmp files
3. Textboxes allowing user to manipulate initial conditions of simulation
4. Button that starts the simulation
5. Reset button restores clear canvas (6) and refreshes its size
6. Canvas, the place where result of simulation is being printed after pressing ‘run’ button

- **‘File’ submenu and import/export of files**

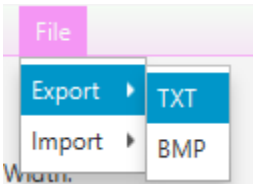

Image 2. Import/export submenu

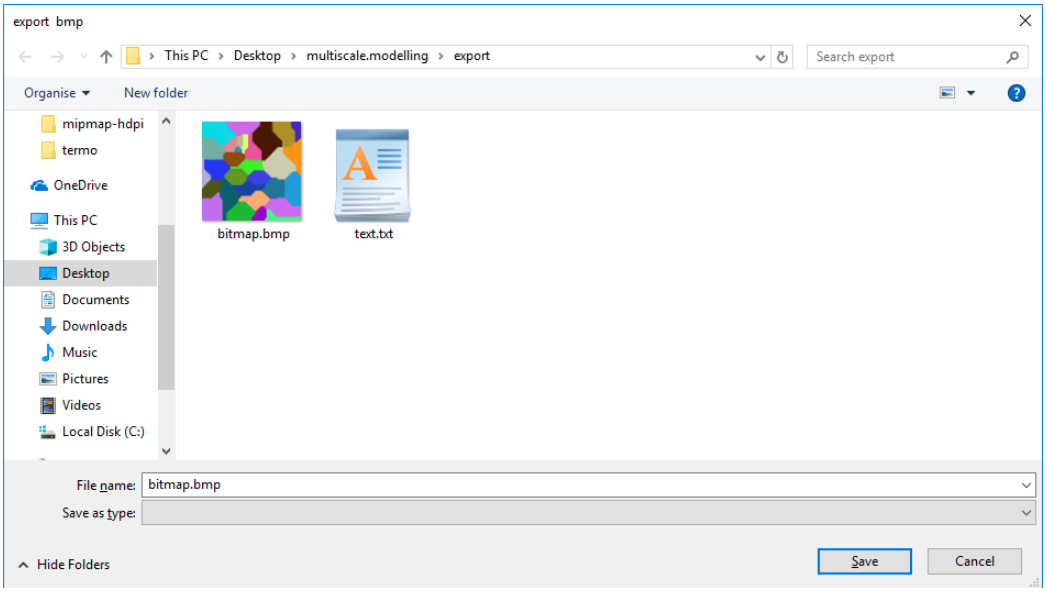

Image 3. This window is displayed whenever user wants to import or export a file.

Submenu ‘File’ contains options of importing and exporting grids in both .txt and .bmp files. After choosing either import or export, new window with path selection pops up (image 3). Image 4 is an example 300x300 grid imported from text file.

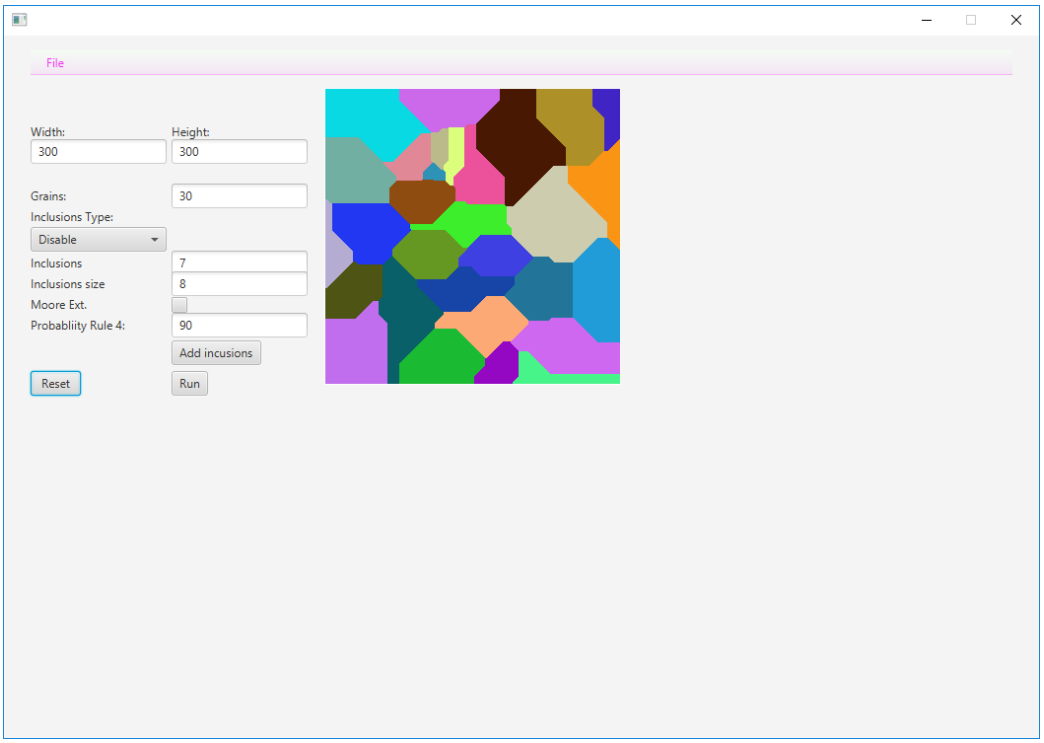

Image 4. Example import from txt file

- **Simulation parameters**

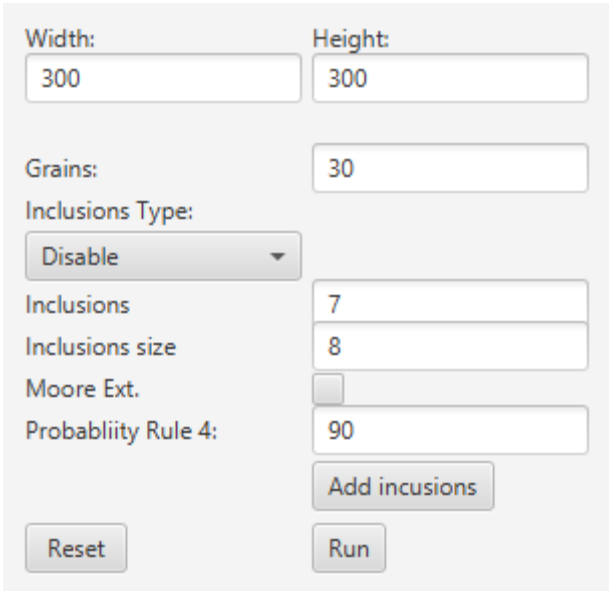

Image 5. Simulation options

In this section user can define dimensions of initial grid and number of grains at the beginning of the simulation. There is also a possibility of adding inclusions, either before starting program (inclusions are spread randomly) or after the simulation, with an option of random location or placing them on the borders of grains. Additionally user can choose between square and round inclusion type. Next two textboxes provide user with a possibility of changing number of added inclusions and their sizes. Examples with inclusions are provided in images below.

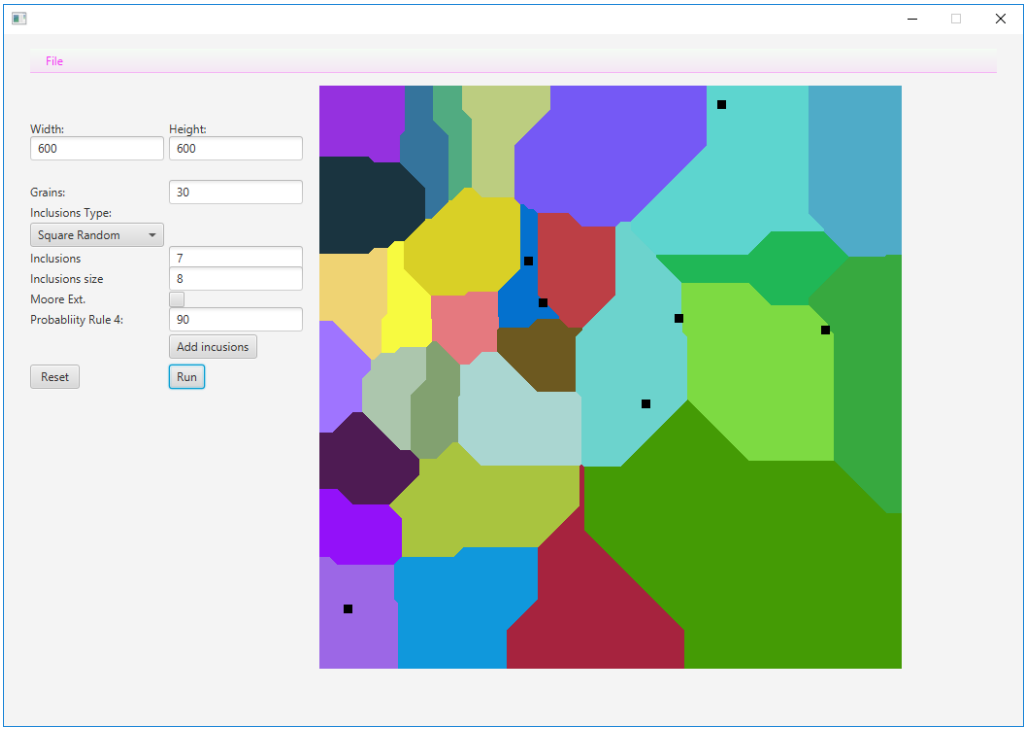

Image 6. Square inclusions added randomly before simulation

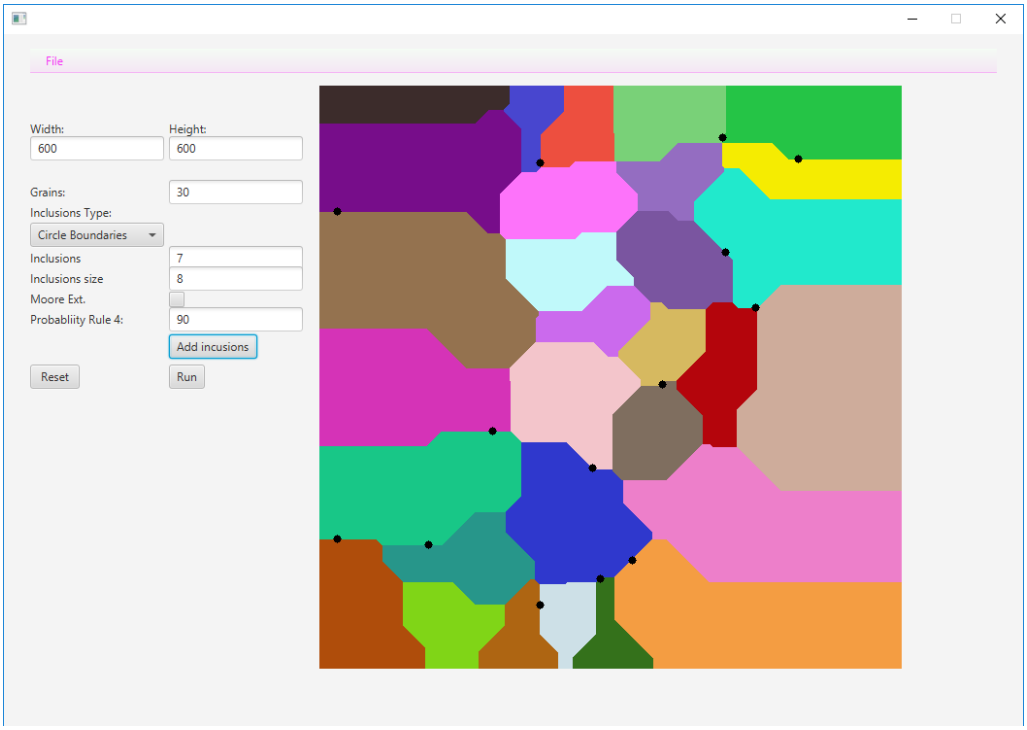

Image 7. Round inclusions added after simulation on grain boundaries

In simulation parameters section it is also possible to select a more advanced method of simulation – Extended Moore which extends process by adding a probabilistic aspect. To enable this option, user has to simply check a 'Moore Ext.' Option, and provide program with a probability of fourth rule (number from 1 to 100). Comparison of two marginal values of probability is shown below.

Image 8. Comparison of Moore with probability equal to 10 % (left) and 90 %(right)

As we can see on provided results, the greater chance for occurrence of 4th rule, the closer the result gets to one provided by simple grain growth algorithm.

4. Comparison with real life microstructure

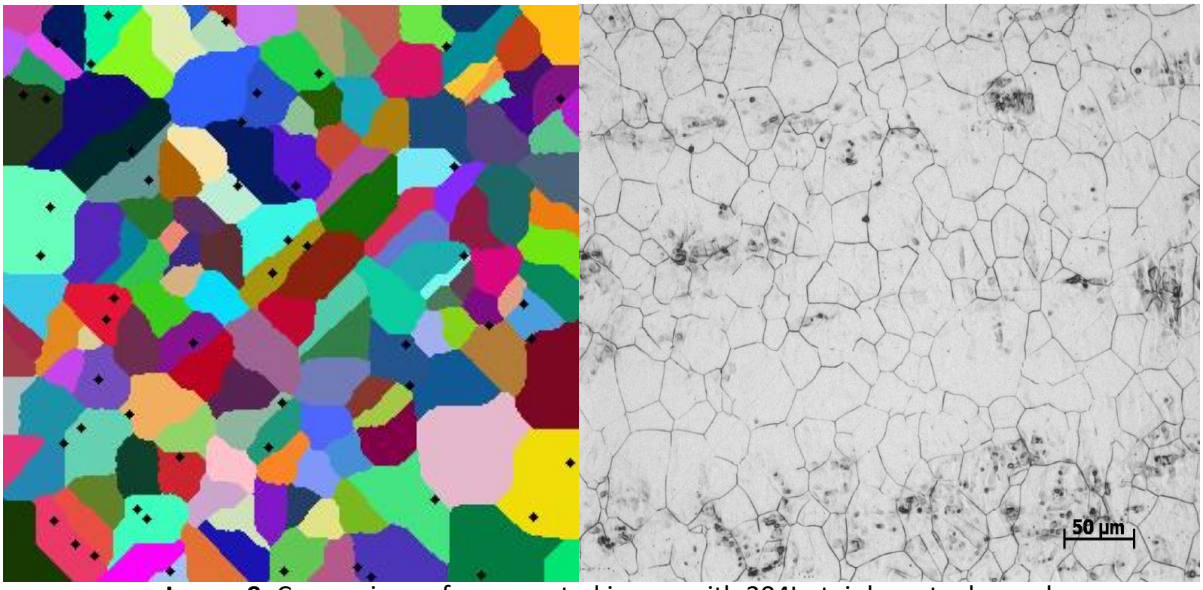

Image 9. Comparison of a generated image with 304L stainless steel sample,
source: https://www.researchgate.net/figure/Optical-image-of-the-microstructure-of-the-etched-304L-stainless-steel-sample_fig1_262936340

Simulation of Extended Moore neighbourhood with 90% probability of 4th rule provides quite satisfactory results in comparison to provided sample.

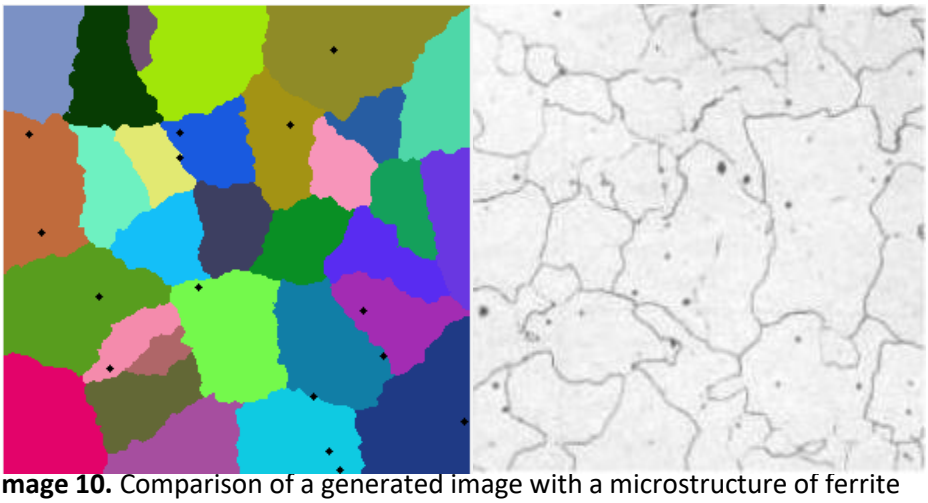

Image 10. Comparison of a generated image with a microstructure of ferrite
Source: http://practicalmaintenance.net/wp-content/uploads/Microstructure-of-Ferrite.jpg

Comparing result for 2% chance for occurrence of 4th rule provides a result which still might be classified as satisfactionary, but there are visible differences in two structures. Generated structure is a little wobbly on grain borders, while real one has more sharp and straight edges.

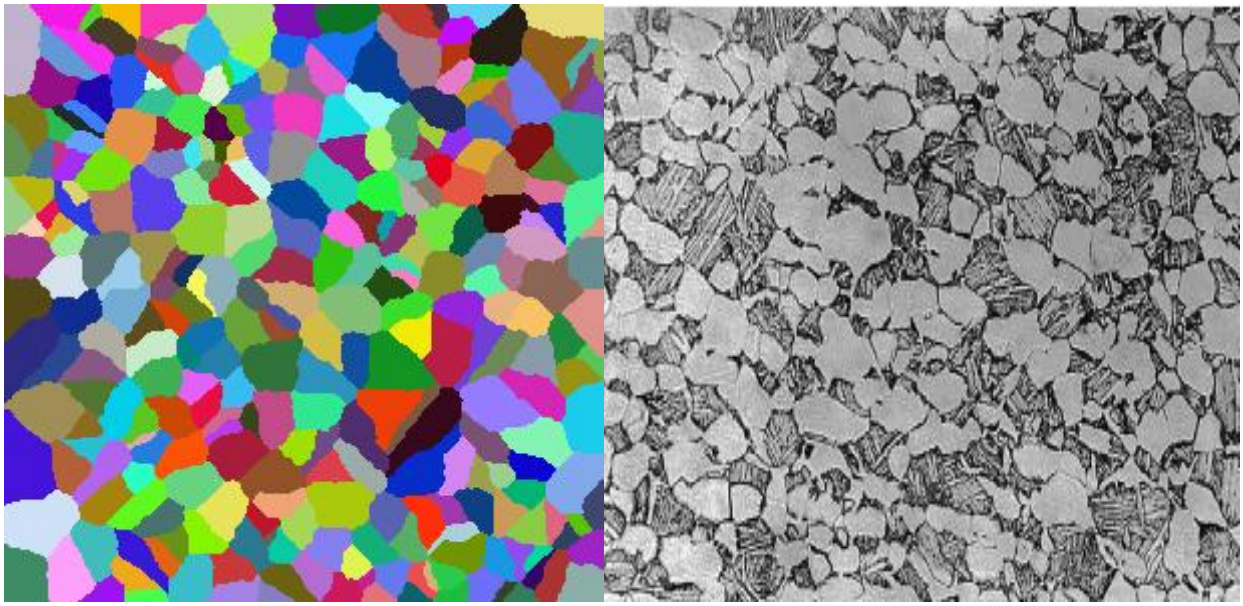

Image 11. Comparison of a generated image with a micrograph of Titanium (Ti-6Al-4V)
Source: https://www2.lbl.gov/ritchie/Library/PDF/MTransLamellar.pdf page 3

Simulating that specific titanium alloy provides user with a satisfying result image, although it is required to mention that many other titanium based alloys are hard-to-impossible to receive through any possible combination of settings in this application.

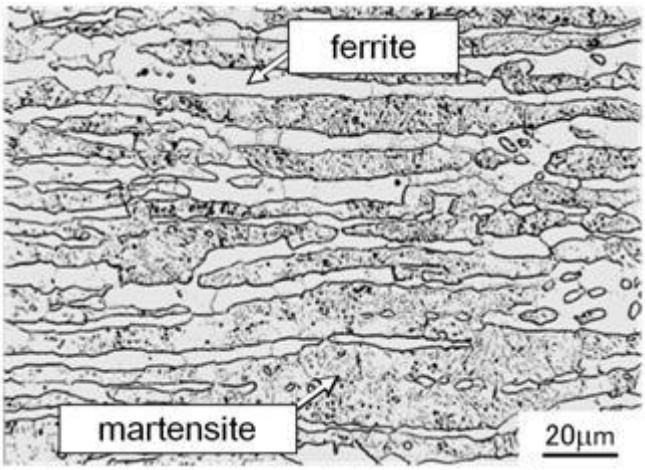

Image 12. Microstructure of super 17Cr Steel
Source: http://www.nssmc.com/en/news/old_smi/2011/images/e-news2011-12-07-01-01.jpg

As a last example, author is obliged to provide at least one example of microstructure which is impossible to simulate in his application. Due to dual phase structure and long grain, martensite – ferrite steel is, to author's belief, impossible or nearly impossible to be simulated. Many other compounds with similar structure exist, that is why this application's uses are limited only to specific materials.

5. Summary

As an overall conclusion, application is capable of simulating real, existent structures, but of course not all of them. For example, in current state of application, due to the problems encountered in the development phase its quite hard to simulate dual phase microstructures because option for selecting and deleting colours of grains is not available. Many real structures still can be generated, but at least some level of knowledge in crystallography is recommended in order to achieve desired effects.

Regarding optimization of the program – author's present skill set has been a major obstacle in achieving better application performance, and mainly because of that the presented product might be developed to use less resources and work faster (for example by using multiple threads).