Przemysław Cich, gr. 1, IS IMiIP, st. 2

Multiscale Modelling – II Report

## 1. User Interface

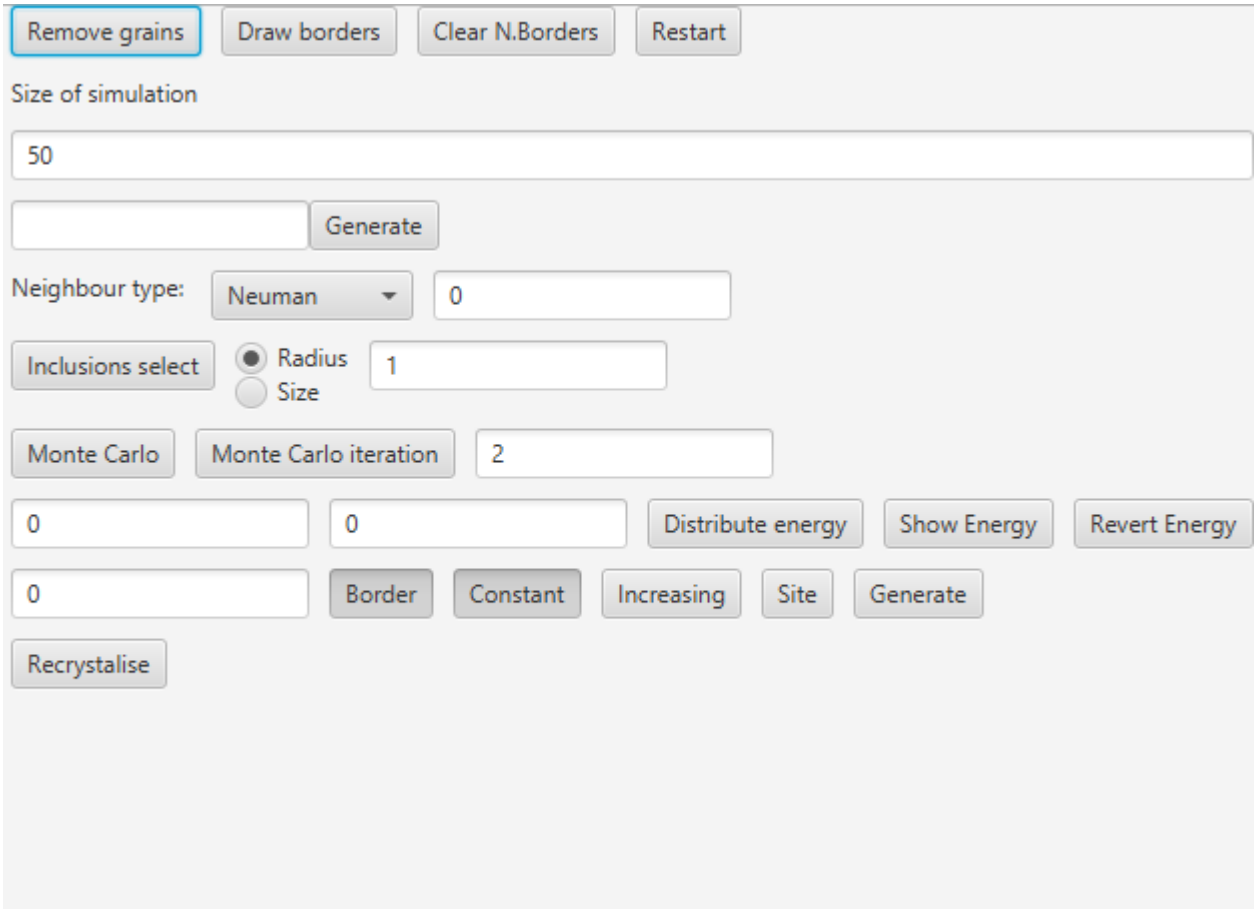



*Illustration 1: User interface*

Application consist of two major part. Working area, which allows to create and display grains and their growth, and control menu. Menu allows user to adjust many parameters of the grain growth simulation. The most important ones are: size of the simulation, automatic generation of grains, inclusion generation. The application  is also able to save and load microstructure from and to bpm or json file via top menu.

Detailed UI description:

- Top buttons ( Reset, Draw borders ) - applies global changes to the workarea ( reset, substructure etc )
- Size of simulation textfield – specify simulation size
- Neighbourhood type – choice between curvature or moore grain growth
- Generate button and textfield – generates given nyumber of grain

- Inclusion select – allows to change select mode to inclusion ( clicking on workarea will generate inclusion)
- Monte Carlo – allows to generate initial microstructure and then perform MC growth
- Energy - adds and displays energy on grains
- Nucleons – generates first recrystallised grains ( Site, Constant adding or increasingly adding )
- Recrystallization - performs recrystallization

Least, clicking on the work area with left-mouse button will generate single grain, clicking on exsisting grain will mark it for substructure generation and right click will merge it to single grain.

## 2. Technology

Project was written in Kotlin Programming Language with the use of TornadoFx library ( Kotlin dsl port for javaFX). Technology was chosen on personal preferences and desire to learn new skills.

## 3. Scenarios

1. Firs scenario: grain growth using MC method ( 2 different grains )

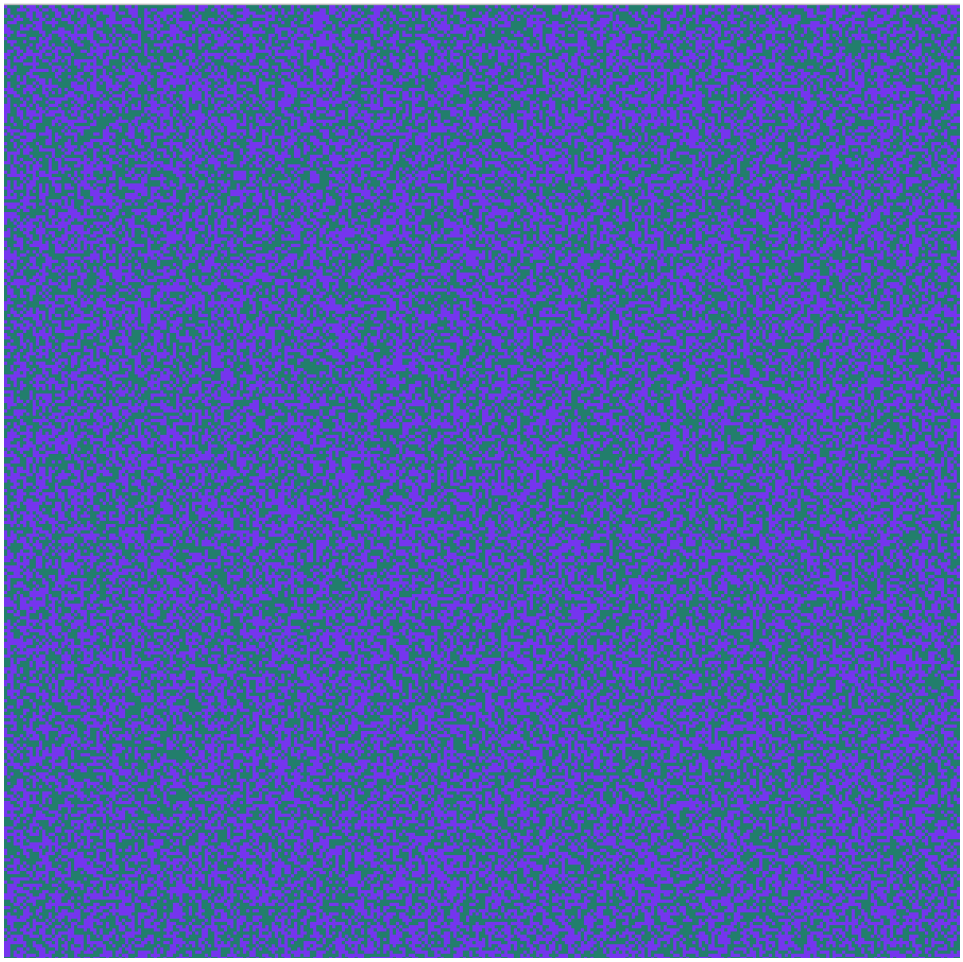


*Illustration 2: Micostructure before MC*

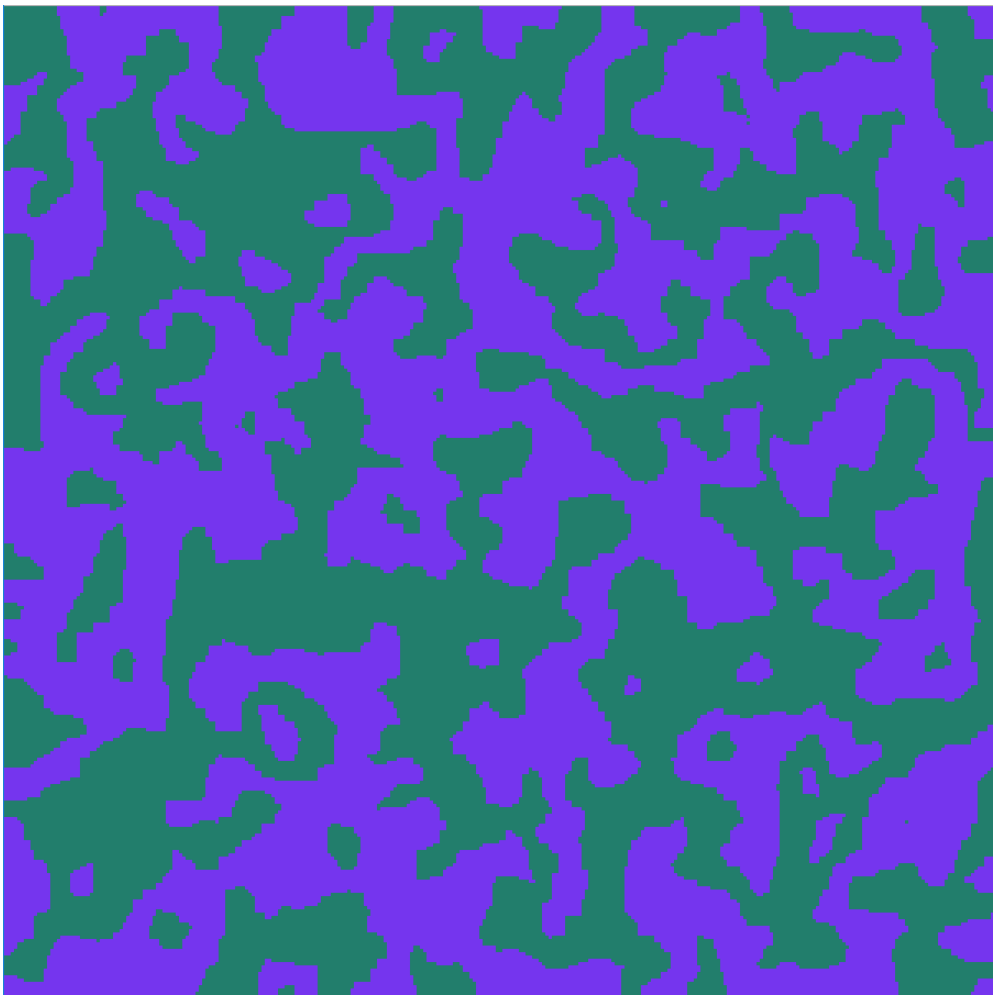


*Illustration 3: Microstructure after MC*

Performing growth with MC method allowed to create very realistic models of microstructure. CA method, because of it's neighbourhood characteristics, had more sharp, unnatural edges. Mc on the other hands is much more smooth.

2. Second scenario: Dual phase structure MC → CA ( 2 grains MC, 7 grains CA )

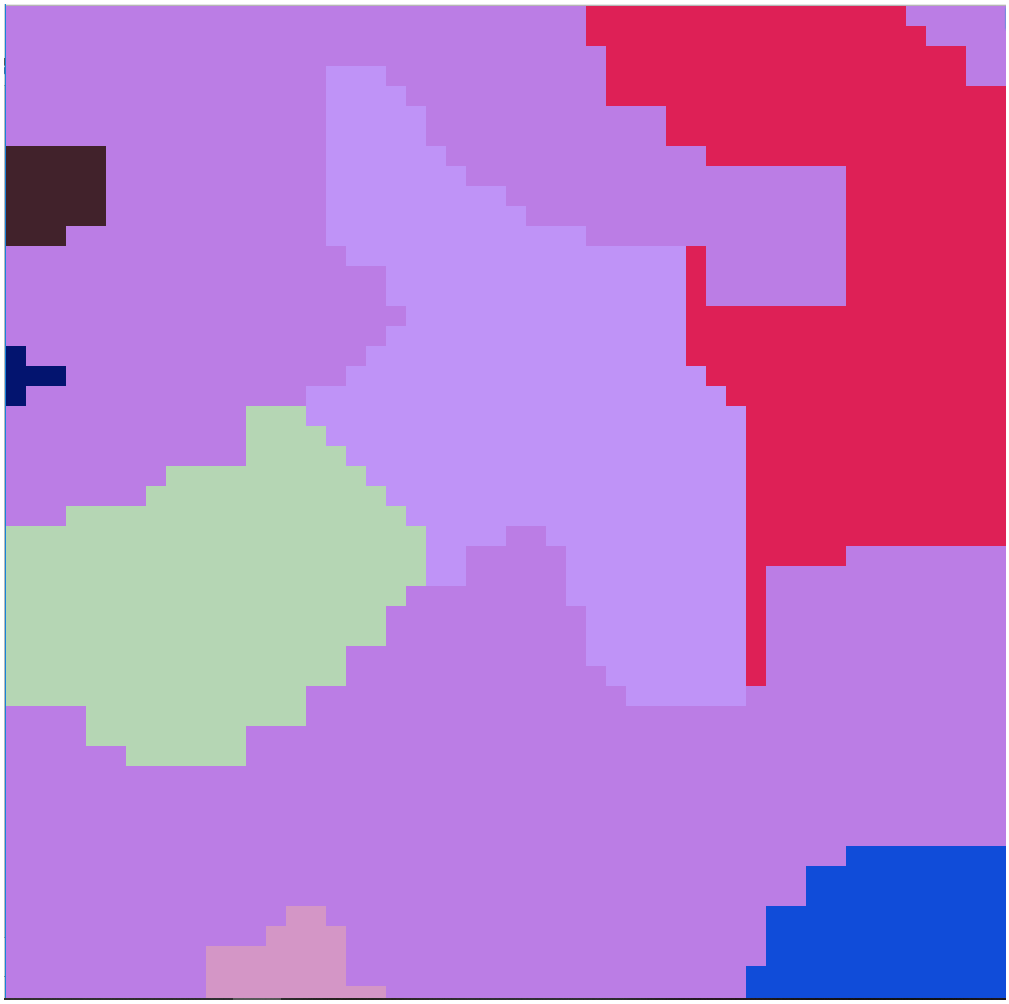


*Illustration 4: MC -> CA Dualphase*

Using dual phase functionality it is possible to create more complex structure. We can model materials with different attributes that are better suited for our needs. In this example MC simulation was performed (with 2 grains ) and then one of the grains was replace with new grains that was enlarged using CA algorithm. It allowed to create above structure.

3. MC structure → Distributing energy → Nucleation and growth

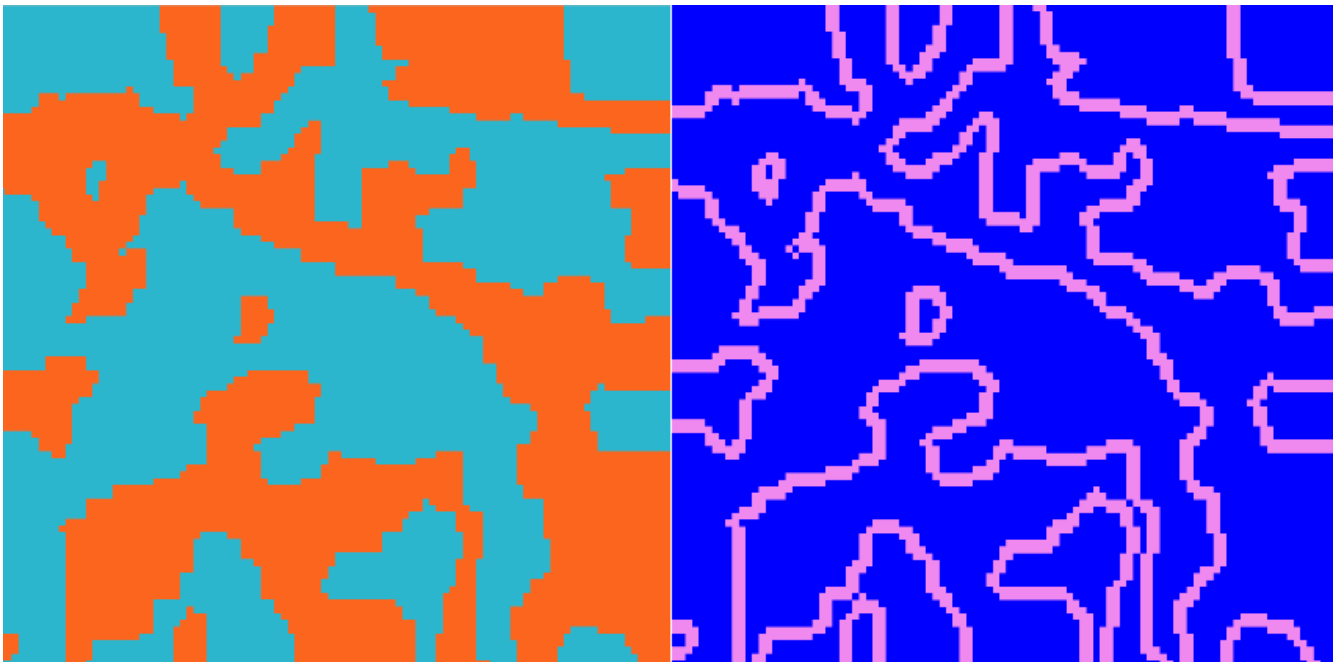


*Illustration 5: MC structure with energy distribution*

First performed MC simulation. After that energy was distribute ( heterogeneous 1 , 12 ) and its distribution was displayed.

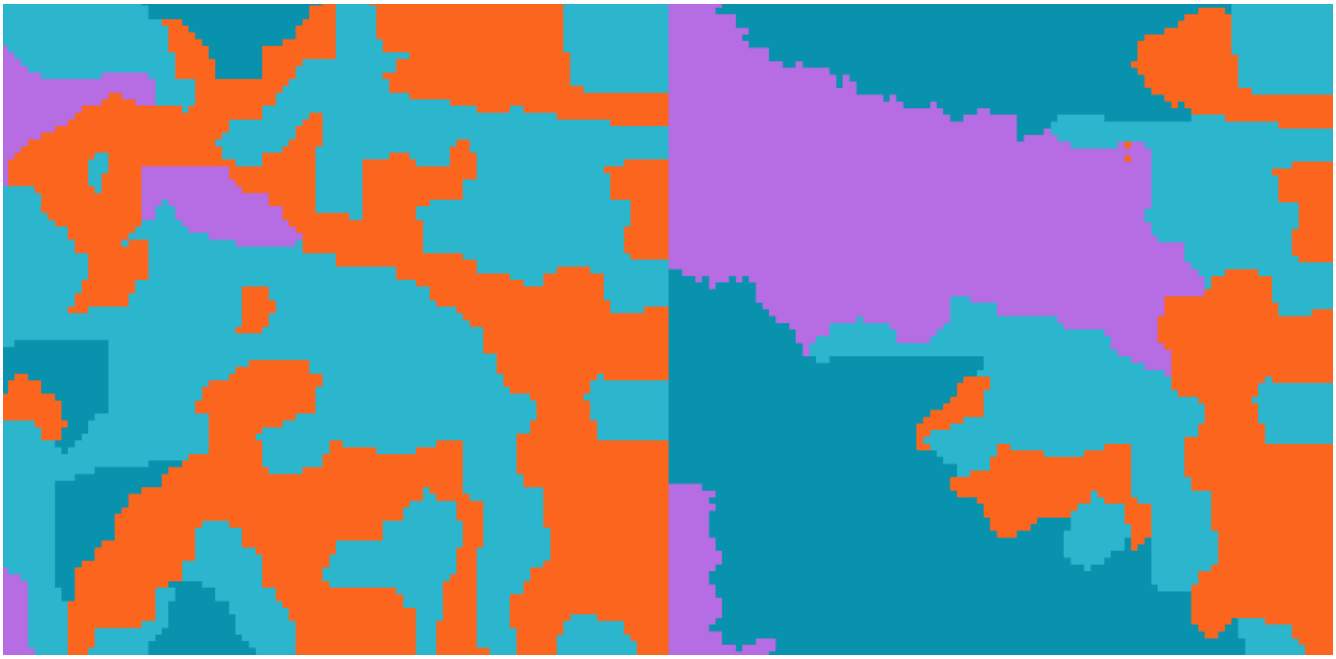


*Illustration 6: Recrystaliztion 9 and 20 iteration*

After nucleation phase ( 5 two-colorded grains )  recrystalization was performed ( 9 and 20 iterations ). In the picture we can clearly see that because of energy distribution the grains are growing along the edges of old ones, and are expanding inside during later iteration ( e.g 20 )

## 4. Real structure comparison

- Single phase
- Aluminium ( 0.1 % Mg )
- Cold rolled → nucleation phase → recrystallization

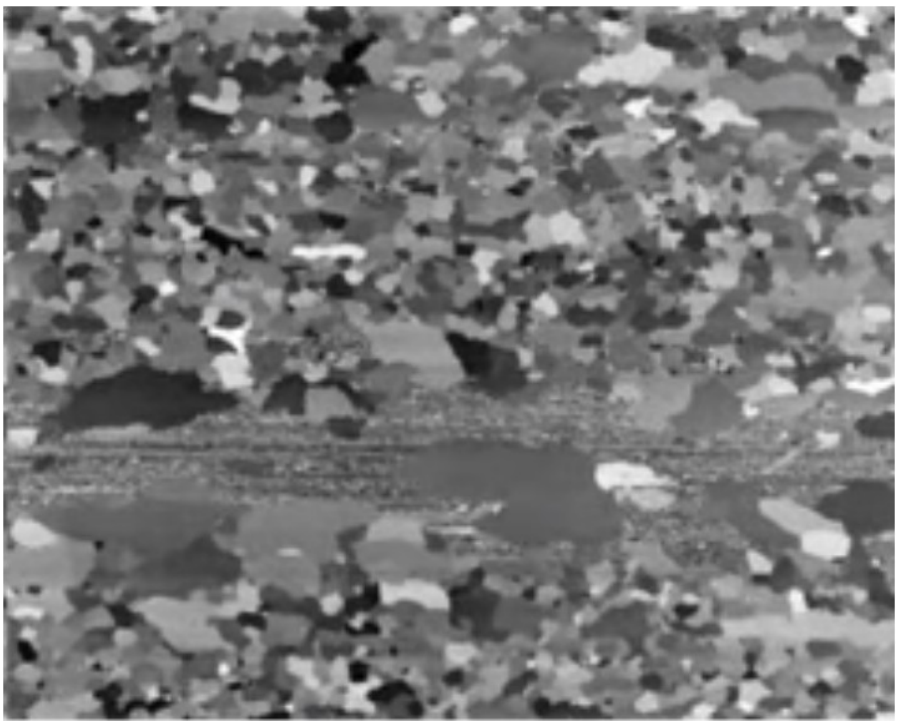
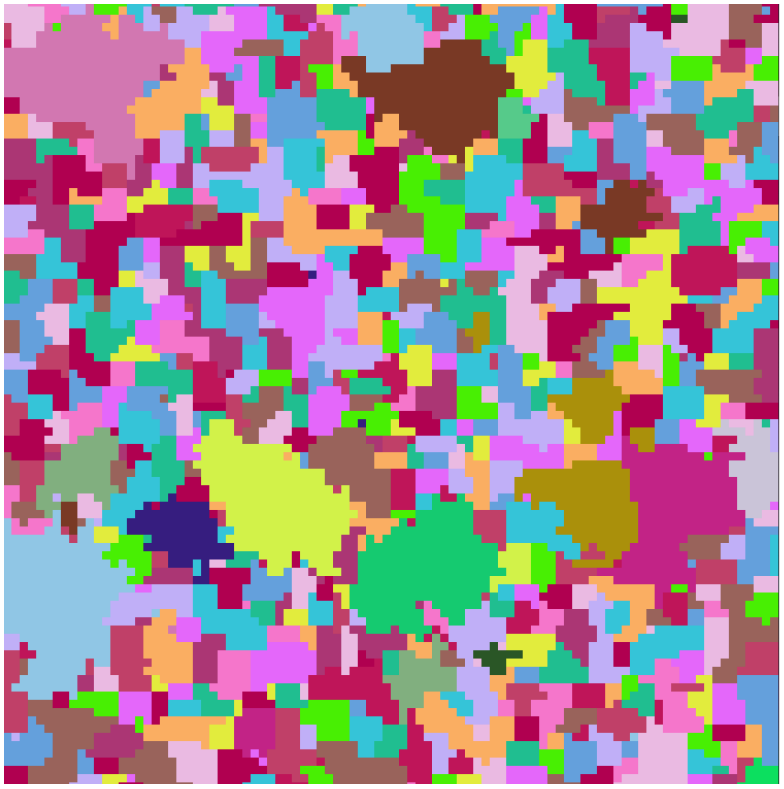


*Illustration 7: Real microstructure*



In this scenario it was much harder to obtain similar structure. It was finally possible with usage of constant adding new nuclei (rate: 5 per iterations). Because of that, the structure diverge in the size of growing ( new ) grains. Some of them are already big, where others are just starting to grow. It was especially rewarding to closely model the middle band of recrystallised grains which needed to be bigger and mostly in one band in comparison with other grains

- High Copper Alloy
- Plate
- Alloy: C17200
- Temper: M20

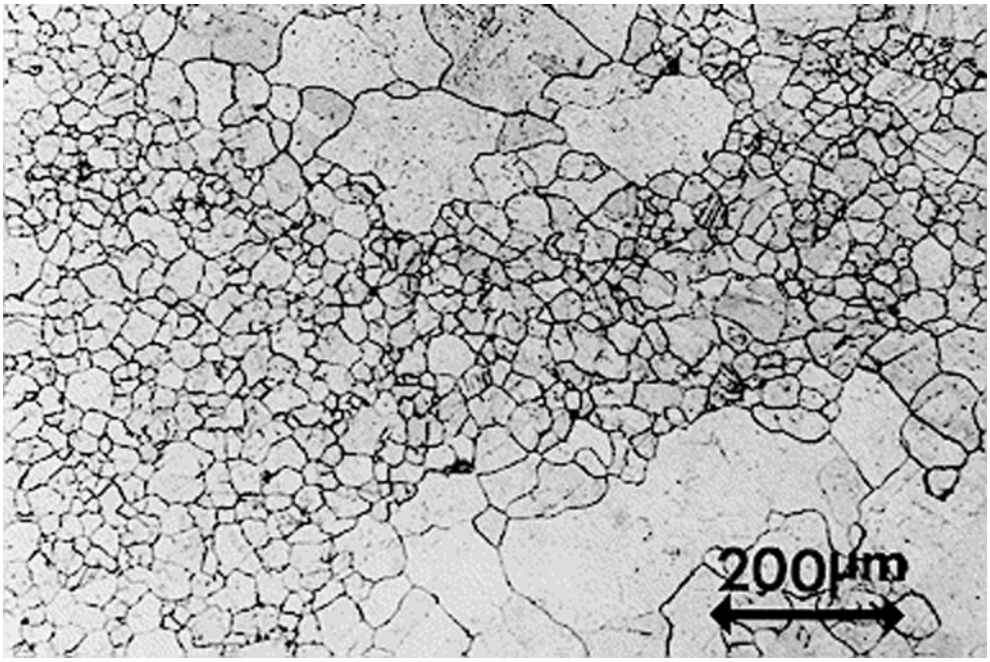



*Figure 1: Real Microstructure*

*Illustration 8: Structure after recrystalization*

In this case the simulation and real microstructure is very similar.  The size and shape of the grains are close enough to the material. To obtains such structure it was necessary to first create CA structure, and then distribute energy and perform recrystallization. The result resemble original microstructure very closely.

## 5. Summary

After testing various configurations and settings, I discovered that program allows to simulate many typical microstructure, it's ui and functionality while minimal allows for free manipulation of structure. Tested scenarios shows that it is fairly easy to generate grains, perform growth, removed some grains and start growth anew. It is also possible to create only borders of grain and simulate grain growth in this domain. Recrystallization functionality combined with MC offers a broad range of additional simulation and modelling tool. With its help it is possible to create more complicate structure that with previous algorithm. Using created software I created structured very similar to real ones, without much of configuration or tweaking ( random nucleation and grain position ). It shows that the algorithm is capable of modelling real life problems. Its main strength comes from the fact that its based on monte carlo algorithm which is choosing next grain in iteration at random. It allows for more smooth edges and more equal distribution of grains. Especially powerful  is combination of CA and MC/recrystallization which allows for much more complex structures.

The performance of the small simulation in acceptable. In low resolution of simulation ( less then 100x100) it is very fast. During this iteration, some performance fixes were applied, and the program has no problem with simulation of size 300x300. The main problem, performance wise, was using immutable collections and mapping over them every time some operations was performed. It created unnecessary resources usage. This issue was resolved and it is now possible to create bigger simulation.