Sylwester Cichocki

Multiscale Modeling – first report

1. Introduction

The goal of the project was to create a program that implements cellular automata and simple grain growth algorithm used to generate microstructures.

2. Programming language

This project was implemented using C# language with Windows Forms framework. C# and Windows Forms library provide tools that are easy to use to create Graphical User Interface. Thanks to this one can easily create GUI and focus on back-end side of application.

3. Graphical User Interface – description

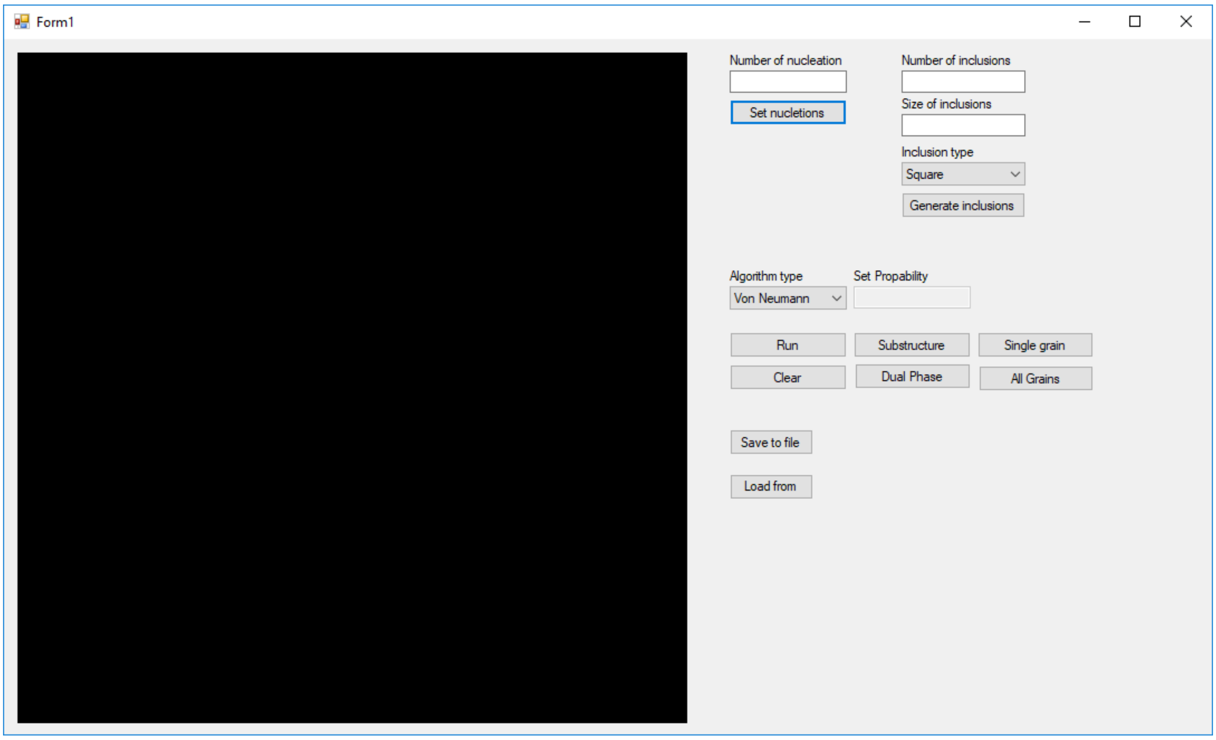

Figure 1 Main window

Figure 1 shows how program looks like. On this picture we can see available configuration options which are described below:

- Number of nucleation – is parameter which determines how many grains will be present in final microstructure.
- Number of inclusions – is optional parameter where you can set number of inclusions present in final microstructure. You can add inclusions before performing grain growth algorithm but also after grain growth. In that case inclusions will be added on grain boundaries. Examples of inclusions are presented on Figure 4.
- Size of inclusion – determines size of inclusions described above.
- Inclusion type – you can choose between square and circular inclusions type.
- Algorithm type – here you can choose type of neighbourhood which is Von Neumann and Advanced Moore. When choosing Advanced moore neighbourhood type the Set Probability text box becomes active where you can type probability. Examples are presented on Figure 2.
- Save and load from files – this two buttons allows user to save or load generated structure to or from file.
- Substructure and Dual Phase – allows user to choose few grains by clicking them and generate substructure or dual phase microstructure.
- Single and All grains – allows user to choose few grains by clicking them or all grains in order to present their grain boundaries.

4. Results

a. **Class 1:** Simple grain growth CA and **Class 4:** extension of Moore neighbourhood

Here is example of two microstructures generated by application. One on the left hand side is generated by using classic von Neumann neighbourhood type, and the other one with advanced moore type. The space is set to 300x300 and you can't change that. There are 100 nucleons in both types.

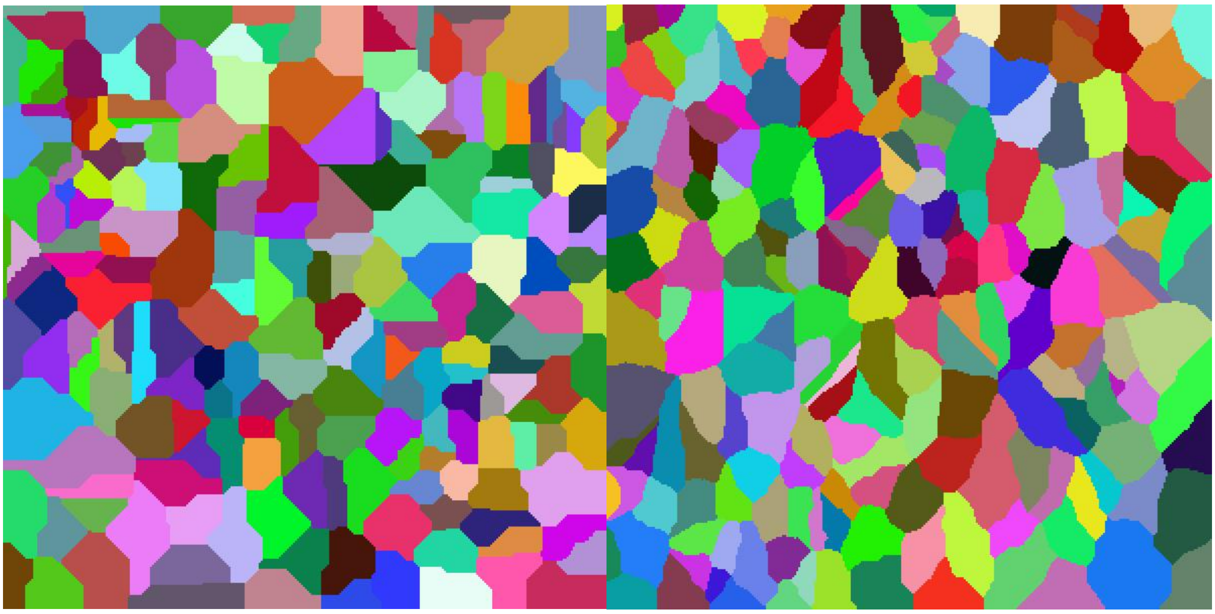

Figure 2 Example of microstructure generated using Von Neumann on left and Advanced Moore with probability 10% on right.

b. **Class 2:** Microstructure import/export: txt/bmp.

In second classes we implemented importing and exporting to files. In my application when user click on Save to file button you can choose where to save your results. Microstructure is saved as bmp file and txt file. You can also load from those two types of files. Example of txt file structure is shown on Figure 3.

```
300 300
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 125
GRAIN 4
GRAIN 4
GRAIN 4
```

Figure 3 Text file structure

c. **Class 3:** Inclusion

Here are some examples of inclusion in app shown on Figure 4. There are two types of inclusions square and circular. Both types can be added before simulation witch results in them being placed in random places and also after simulation then they will be added on grain boundaries. User can specify type, size and number of inclusions.

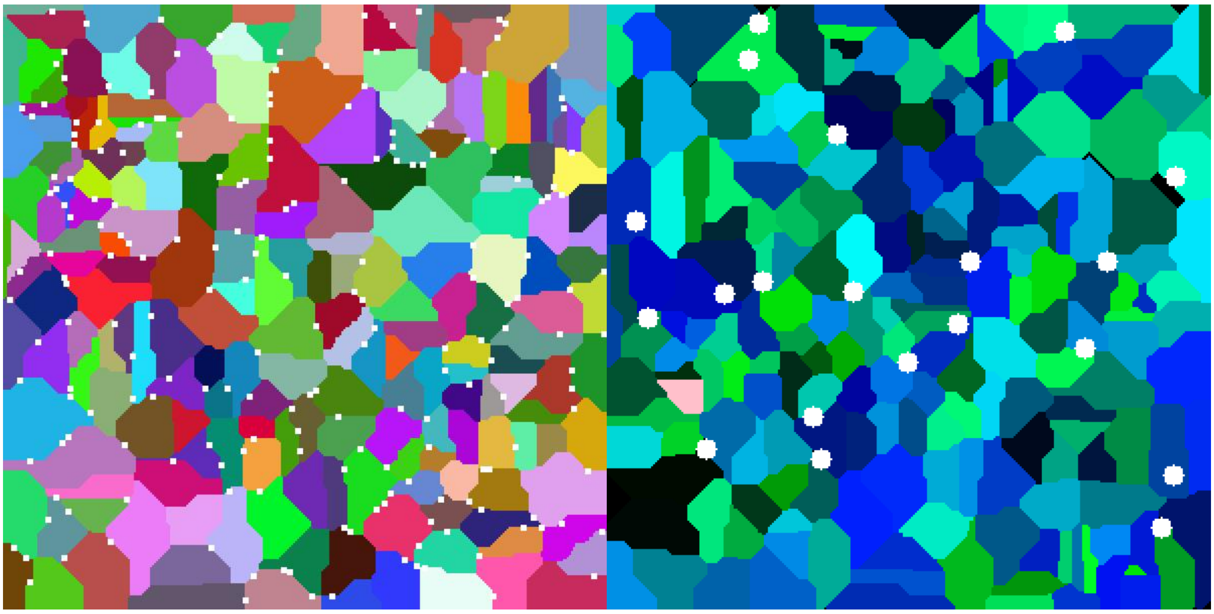

Figure 4 Example of two inclusions type square on left and circle on right.

d. **Class 5:** Different microstructure type: – substructure – dualphase

After initial generation of microstructure user have possibility to pick few grains by clicking them and choose either substructure or dual phase option. When doing that those grains will stay in place during second simulation. Example show on Figure 5.

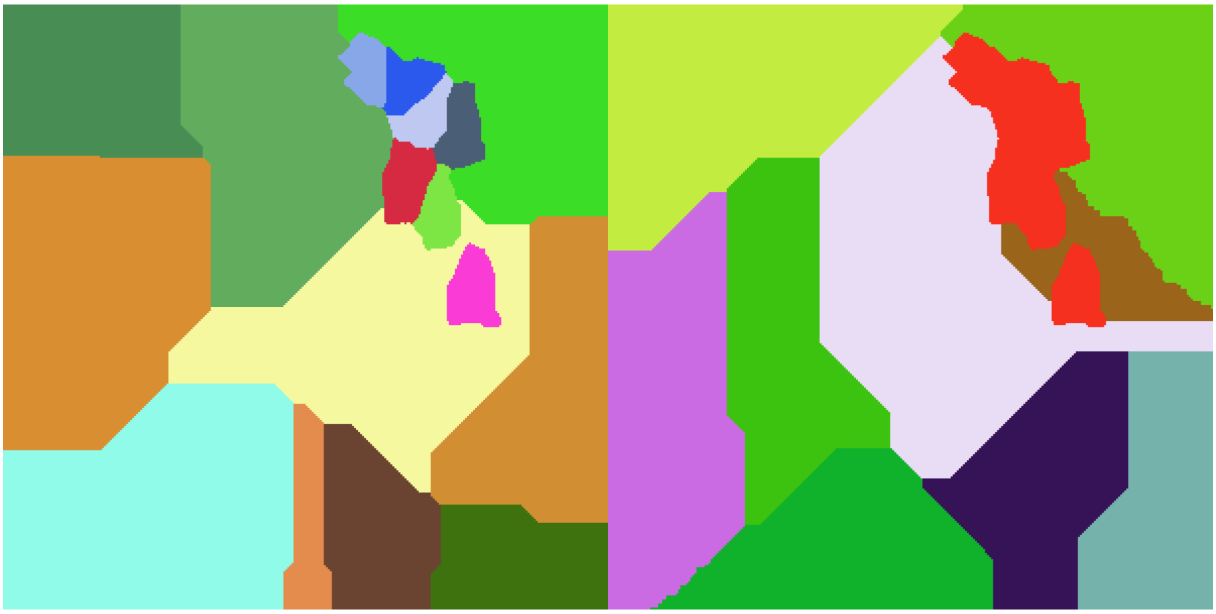

Figure 5 Example of substructure and dual phase.

e. **Class 6:** Grain boundaries selection

After initial generation of microstructure user have possibility to pick few grains by clicking them and click single grain button. That action will result in clearing entire space and leaving boundaries of grains picked by user. User also can click on All grains button which will result in leaving boundaries of all grain on screen. After that user can perform another simulation. Grains will be treated as inclusions. Example of that kind of action is presented on Figure 6.

Figure 6 Grain boundaries and microstructure generated in them.

5. Comparison of real microstructures with microstructures generated with program

On Figure 7 shows comparison of microstructure of polycrystalline metal with microstructure generated with app. They are similar you can see big grains and few smaller grains and also grain boundaries.

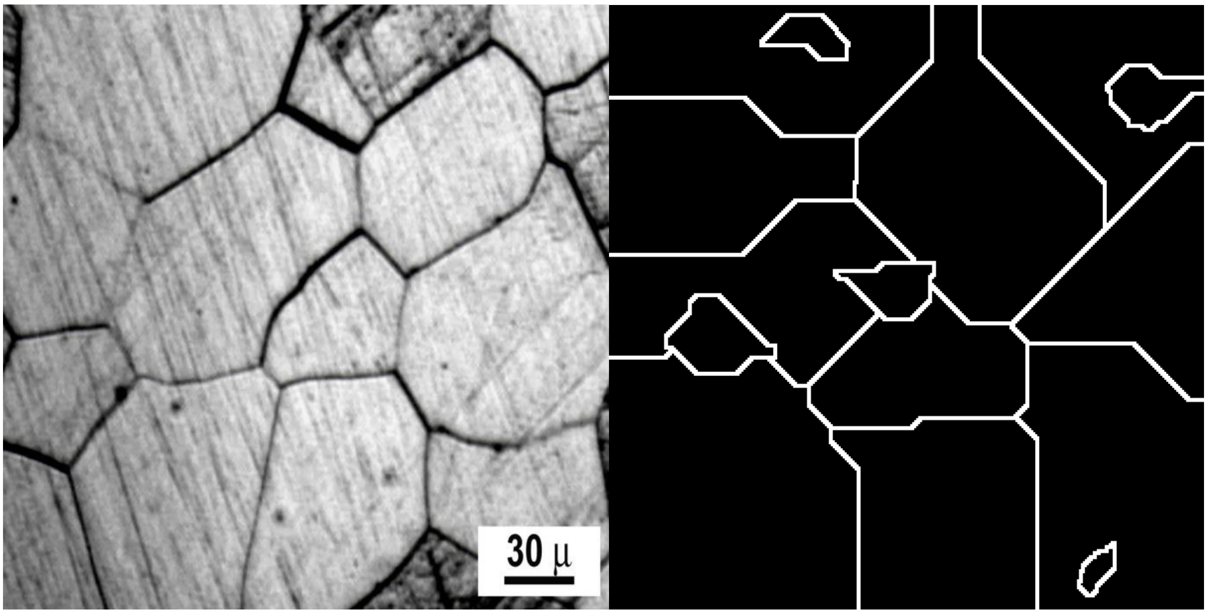

Figure 7 Micrograph of a polycrystalline metal with structure generated in app. Source: https://en.wikipedia.org/wiki/Grain_boundary

On Figure 8 is presented comparison with microstructure of cold rolled steel with one generated in app. To achieve similar microstructure advanced Moore with probability 10% was used and also around 200 square inclusions.

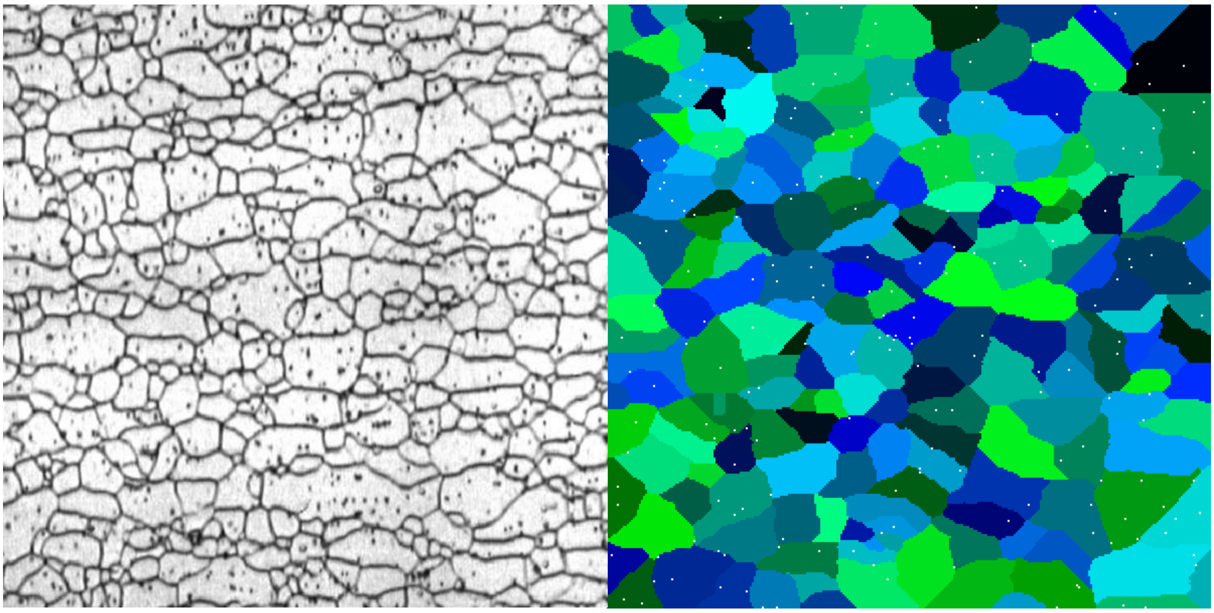

Figure 8 The microstructure of an HSLA340 cold rolled steel with structure generated in app. Source: https://scientific-python.readthedocs.io/en/latest/notebooks_rst/5_Image_Processing/04_Exercices/08_Image_Processing_Practical_Work.html

Here on Figure 9 is example of dual phase steel generated in application. In first step one need to create initial microstructure. Then select roughly half grains and select dual phase option. In last step you generate final microstructure.

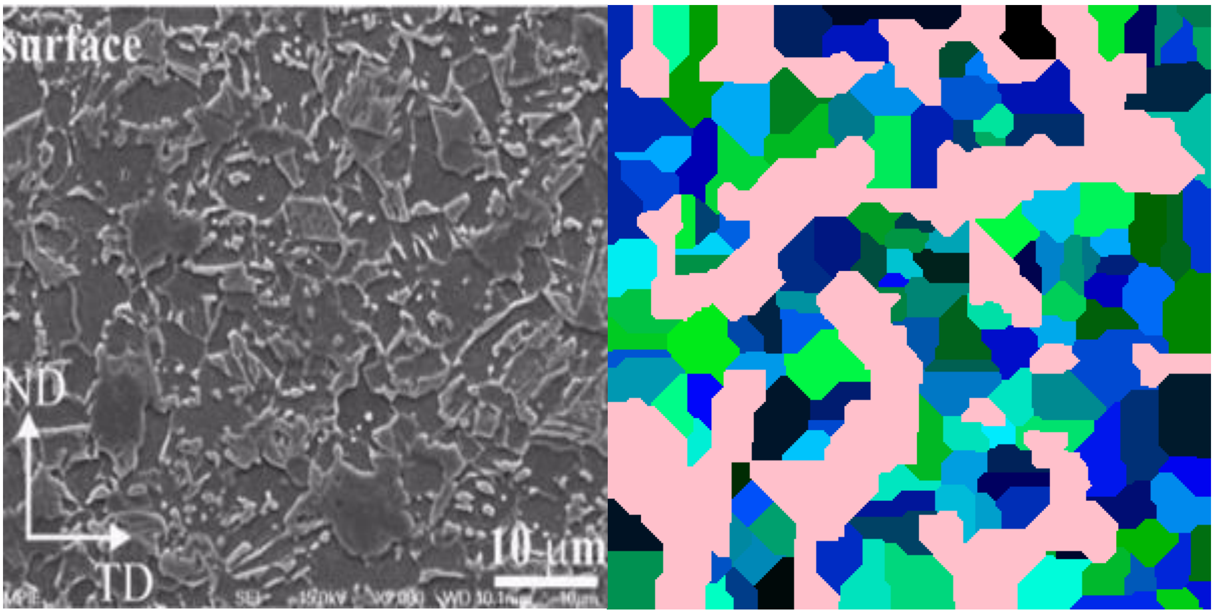

Figure 9 Microstructure of Dual Phase Steel with structure generated in app. Source: http://www.dierk-raabe.com/dual-phase-steels/

6. Conclusions

After looking at results presented on Figures 7,8,9 you can see that achieving microstructures that is similar to the real ones is possible but it requires time and manipulation of options such as number of nucleons which determines final size of grains. There is also problem with final grains shape which depends on used neighbourhood type.

In general cellular automaton with simple grain growth algorithm can be successfully used to generate microstructures that are very similar to the real ones. Some will require additional methods such as Advanced Moore neighbourhood but still it’s a good tool.