Multiscale Modelling

Report I

Mateusz Mazur

1. Technology

The app is written in C#, which is multi-paradigm programming language encompassing strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented and component-oriented programming disciplines. It was developed around 2000 by Microsoft as part of its .NET initiative. C# is one of the programming languages designed for the Common Language Infrastructure. My app use Windows Presentation Foundation (WPF) which is a graphical subsystem originally developed by Microsoft for rendering user interfaces in Windows-based applications. WPF, previously known as "Avalon", was initially released as part of .NET Framework 3.0 in 2006. WPF uses DirectX and attempts to provide a consistent programming model for building applications. It separates the user interface from business logic, and resembles similar XML-oriented object models, such as those implemented in XUL and SVG.

2. Programme features

This chapter will discuss the main functionalities of the application.

2.1. GUI

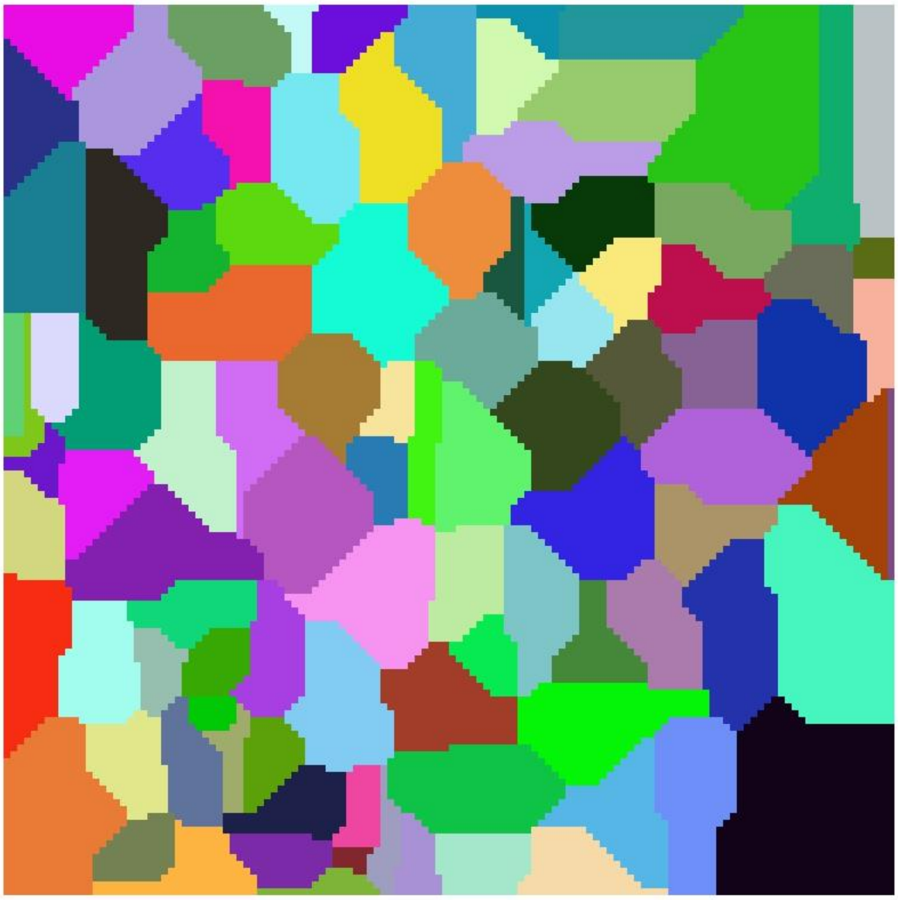
The graphical interface of the application is shown in the picture 1. It consists of buttons and fields enabling the user to enter data located. They are located at the left side of the window. The right part of the window is occupied by a canvas on which the result of the simulation will be presented.

Picture 1. Application GUI

Upper fields such as **Grid width**, **Grid height** allows user to set a size of the calculation area. **Grains amount** field is used to set a number of unique grains. The application will place the grains in a random position inside the grid. In order to add grains, user have to press the **Draw grid** button. To add grains after some of them have been already drawn, user can use **Add to Grid** button and place number of additional grains in **How many add** label. To clear entire grid user have to press **Clear grid** button.

2.2. Simple grain growth

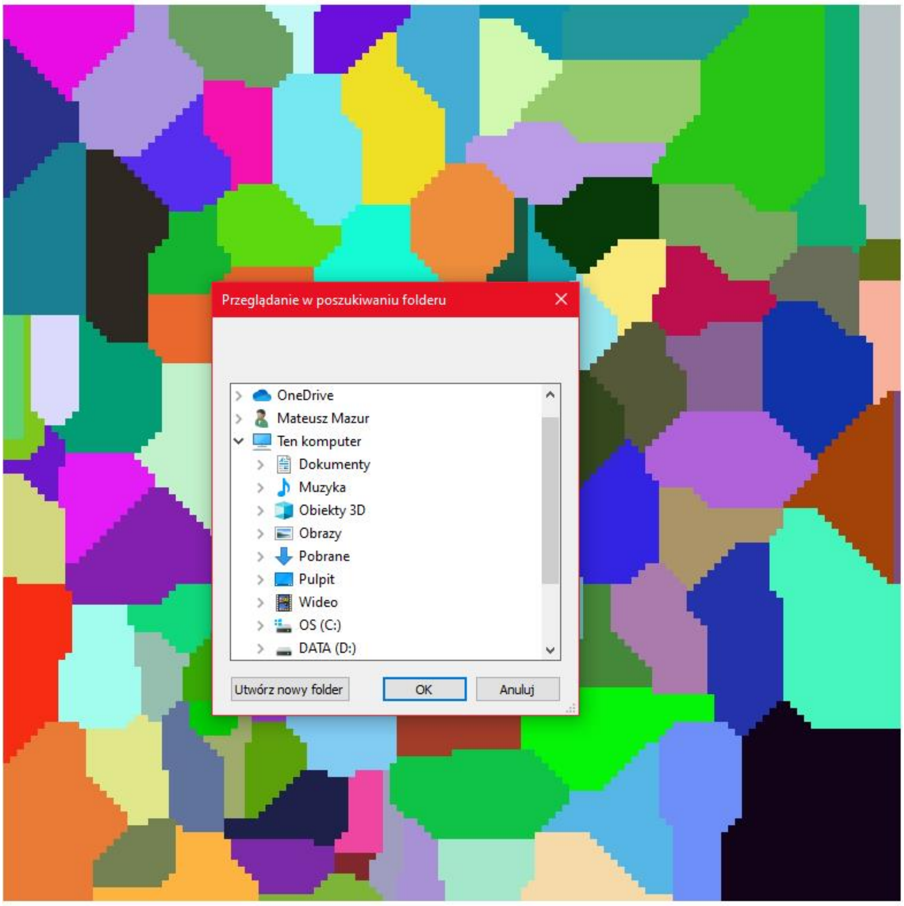
Second app feature is simple grain growth which is shown in the picture 2. To start the grain growth using the von Neumann neighbourhood, the user must select the **Von Neumann** options from the drop down list and press the **Start animation** button. The algorithm will search for the white grain and then check it upper, left, bottom and right neighbour. Current grain will get ID which is the most common in its neighbourhood.



Picture 2. Grain growth

2.3. Microstructure import/export

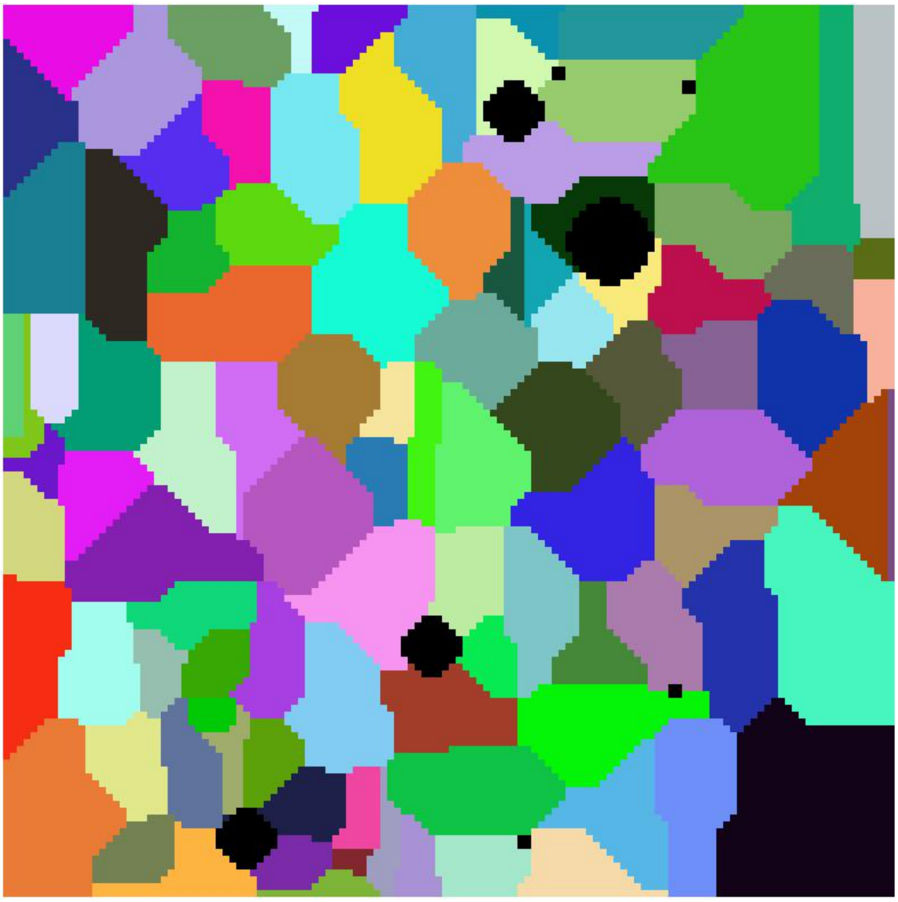
Third app feature is microstructure import/export which is shown in the picture 3. The user has the possibility to save the obtained structure to a .bmp or to a .txt file. Using pop up window he can choose file localization. Using **Input File Name** and **Output File Name** the user can specify name of the file which user want to load or save. To save/load file user have to use **Save File txt, Load File txt, Save File bmp, Load File bmp** buttons.



Picture 3 Saving file in chosen location

2.4. Inclusions

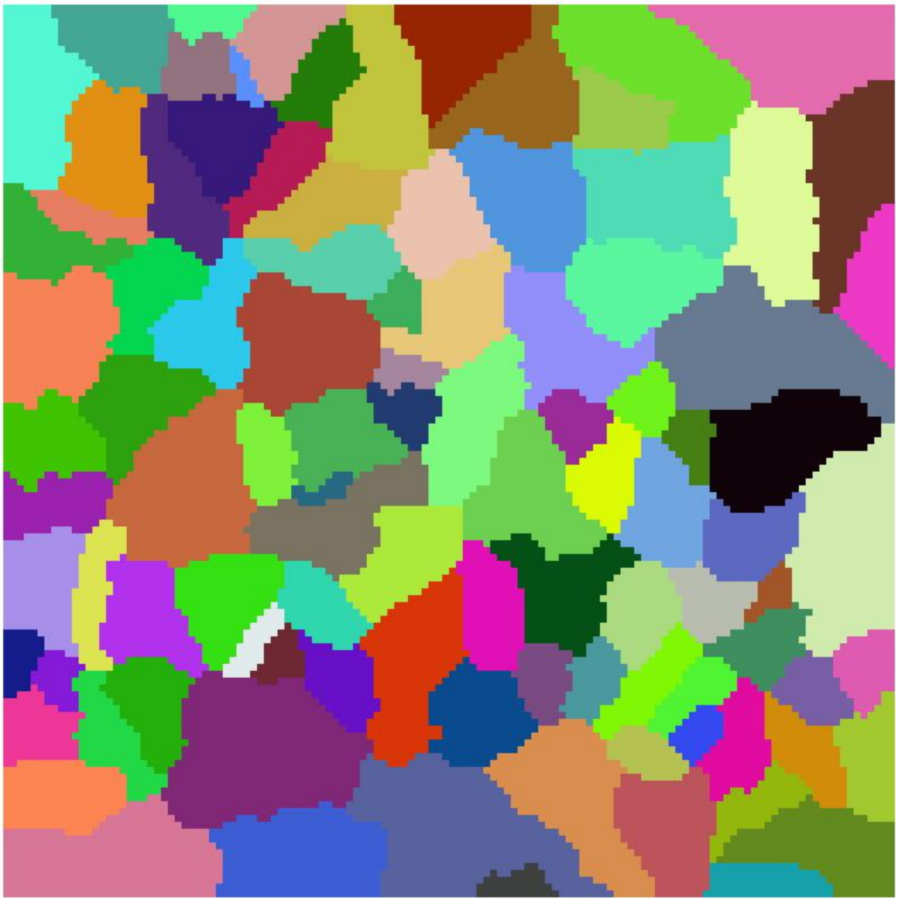
Fourth app feature is ability to add inclusions which is shown in the picture 4. Using drop down list user can choose which type of inclusion will be added to grid. The user can choose between **Square** and **Radial** inclusions. To specify size and amount of inclusions user type appropriate value in **Inclusion amount** and **Inclusion size** text box. Inclusions can be added before and after nucleation. To choose right options user have to check **On grain bounds** check box. The algorithm between drawing an inclusion check if there is enough space for it. To draw radial inclusions the app use Bresenham's algorithm. To check if current position is located on grain bounds, the app check neighbours of current grain. If one of the neighbours have different ID, then condition is fulfilled and inclusion can be added. Inclusions also can't growth.



Picture 4 Addition of inclusions

2.5. Moore 2

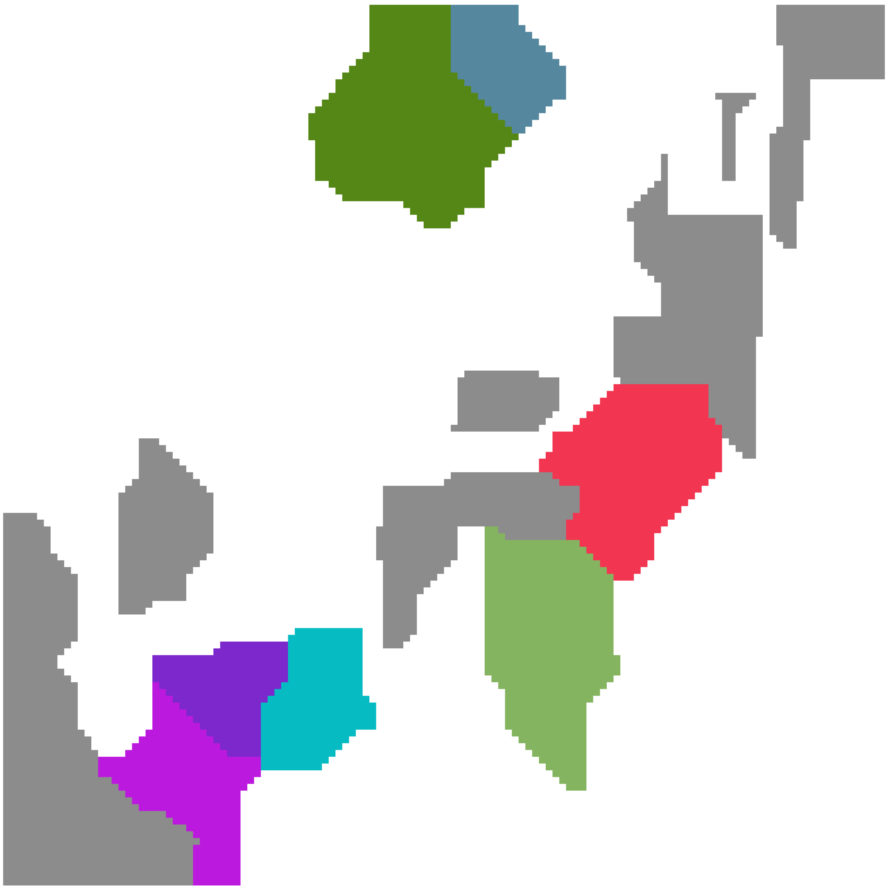
Fifth app feature is grain growth using Moore neighbourhood which is shown in the picture 5. To start the grain growth using the von Neumann neighbourhood, the user must select the **Von Neumann** options from the drop down list and press the **Start animation** button. Before that user have to specify probability for the fourth Moore rule. It can be done with **Moore probability** text box. In Moore neighbourhood the algorithm checks the next rules. If the condition of a given rule is met, the algorithm omits subsequent rules for a given grain. In the first rule The id of particular cell depends on its all neighbours. If five to eight of the cells neighbours id's is equal to S, then cell transforms to the state S. In the second rule the id of particular cell depends on its nearest neighbours. If three of the cells neighbours id's is equal to S, then cell transforms to the state S. In the third rule the id of particular cell depends on its further neighbours. If three of the cells neighbours id's is equal to S, then cell transforms to the state S. In the fourth rule the id of particular cell depends on its all neighbours, and has given % probability chance to change.



Picture 5 Moore growth

2.6. CA -> CA

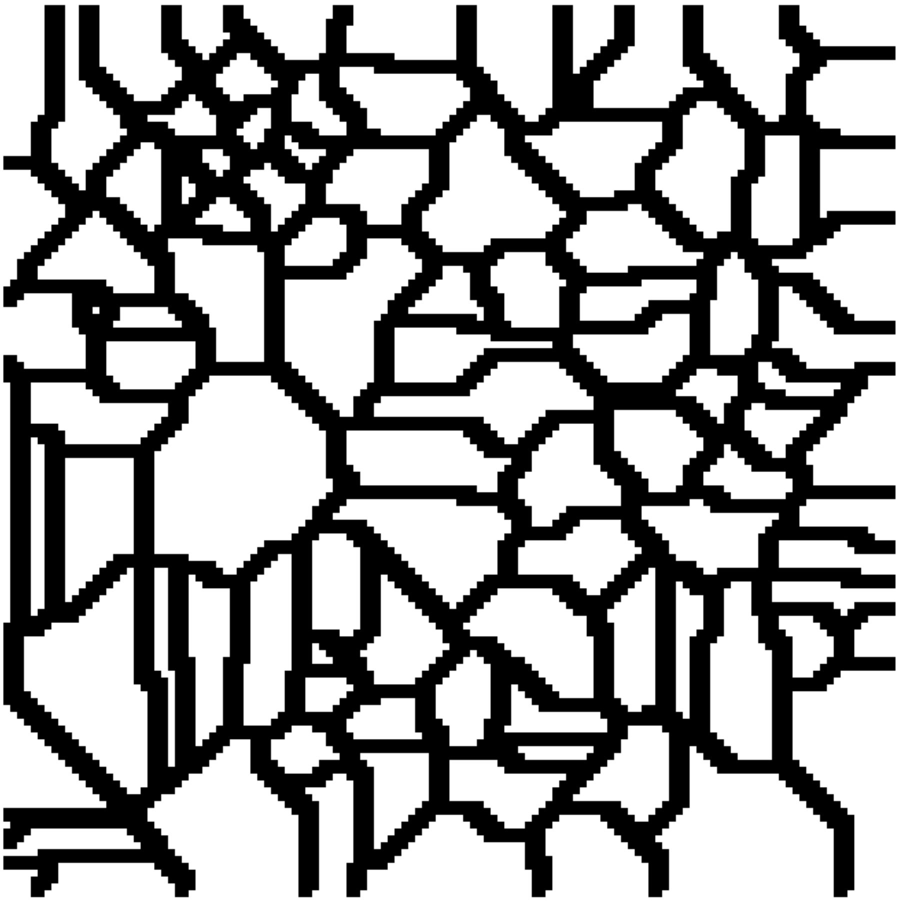
Sixth app feature is generating dual phase or substructure material microstructure which is shown in the picture 6. At this stage, the user can use mouse clicks to select the grain that will be used to simulate dual phase or substructure material properties. **Clear grains To DP** button can be used to clear selected grains if user made a mistake. **Select all grains To DP** button is used to confirm user choice. Then user can erase other grains with **Erase No DP Grains** button. To choose between Dual phase and substructure user can use **Is Sub** check box. Selected grains can’t growth. The user can switch dynamically between DP and substructure during the work of application.



Picture 6 View on Dual phase and substructure generated in application

2.7. Grains boundaries

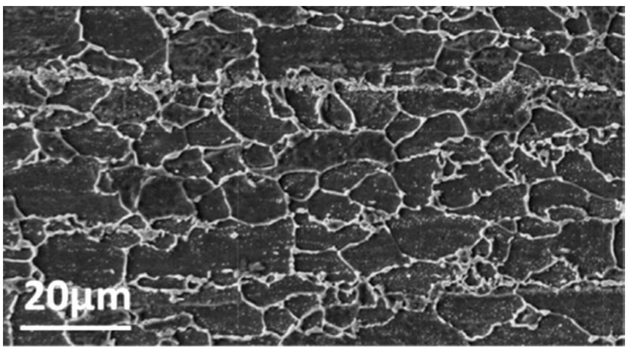
Seventh app feature is adding ground boundaries which is shown in the picture 7. The last functionality allows you to determine the grain boundaries. The user can specify the thickness of the border and choose whether the application should outline all the grains or only those selected by the user with the mouse. The program is **GB %** label displays the ratio of the number of borders to the grid size. The user can choose whether to draw ground boundaries around all grains or only around selected grains using **Draw all ground boundaries** or **Draw SG ground boundaries** buttons. To select grains user can use mouse left click button. Then user can select all grains or only clicked ones with **Select all grains** or **Select not clicked grains button.** After operation user can clear selected grains with **Erase selected grains** button.



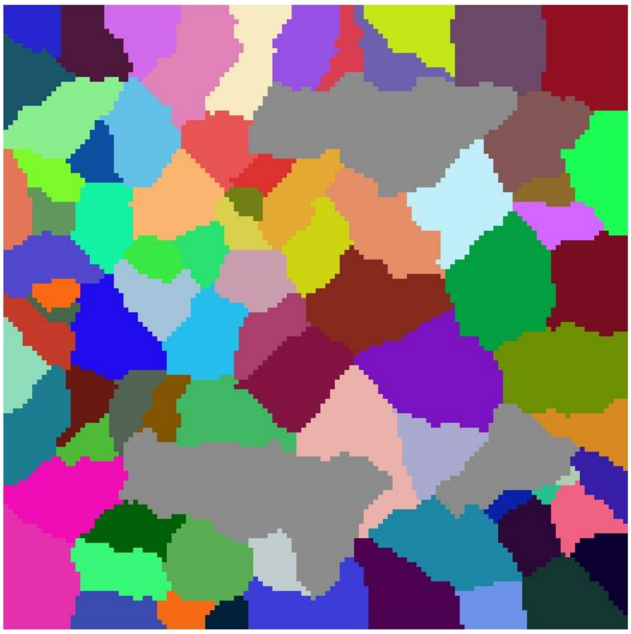
Picture 7 Result of drawing ground boundaries

3. Real microstructures comparison

The picture 8 shows microstructure of a dual-phase steel containing 15% of martensite, mostly located at ferrite grain boundaries [1]. The picture 9 shows the reconstruction of this microstructure achieved with the app.

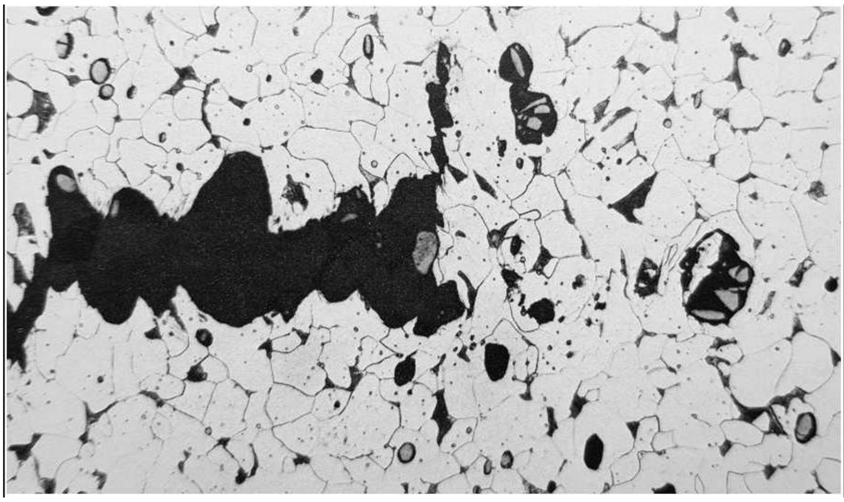


Picture 8 Dual phase steel microstructure

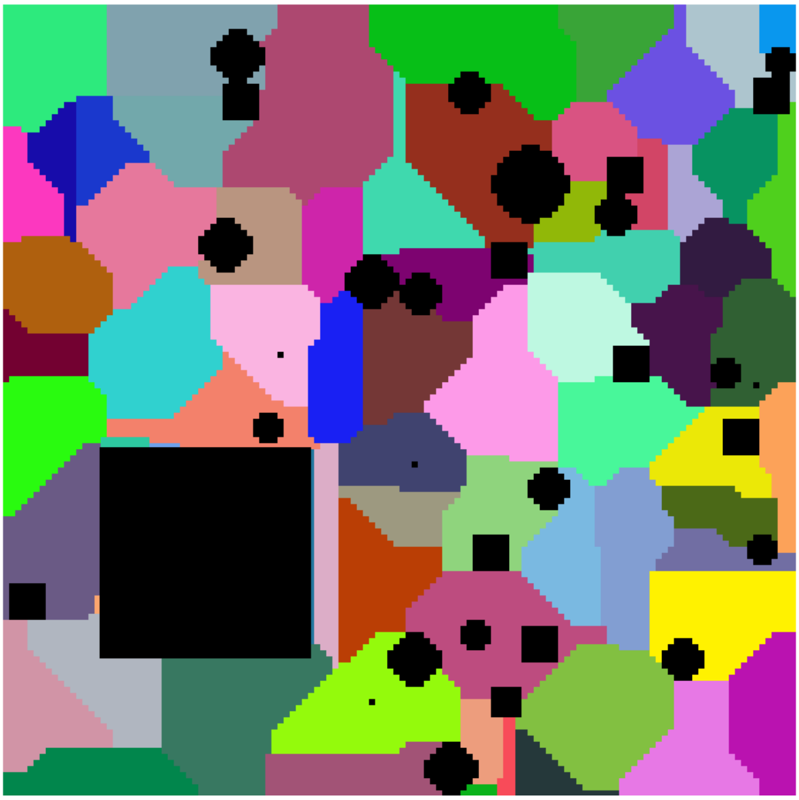


Picture 9 Generated dual phase steel microstructure

The picture 10 shows microstructure with inclusions which was used to check the effects of non-metallic inclusions on properties relevant to the performance of steel in structural and mechanical applications, mostly located at ferrite grain boundaries [2]. The picture 11 shows the reconstruction of this microstructure achieved with the app.

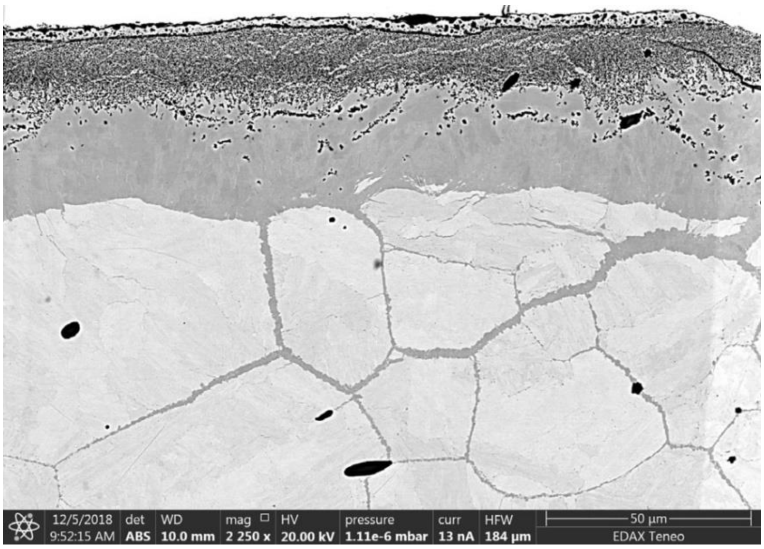


Picture 10 Steel microstructure with inclusions

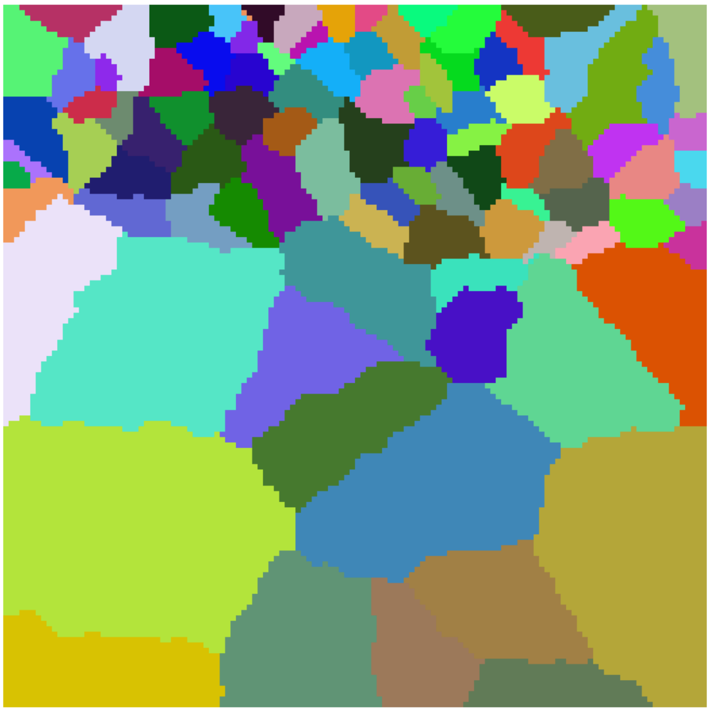


Picture 11 Generated microstructure with inclusions

The picture 12 shows a Backscatter Electron (BSE) image taken from the nitrided surface of a steel sample. The image contrasts primarily show atomic number or density contrast. The brightest region at the top corresponds to the nickel plating used to help with the mechanical polishing and the bright region towards the bottom corresponds to the steel microstructure. [3]. The picture 13 shows the reconstruction of this microstructure achieved with the app.



Picture 12 Microstructure of a nitrided steel



Picture 13 Generated structure similar to nitrided steel

4. Conclusions and summary

In some cases microstructures generated by the app are similar to the real microstructures. Given that this application can be used for simulating simple microstructures with good accuracy. Unfortunately, with the more complex microstructure, the application effectiveness drops.

5. Bibliography

[1] Dual phase steel microstructure https://www.researchgate.net/figure/Microstructure-of-a-dual-phase-steel-containing-15-of-martensite-mostly-located-at_fig4_320104922

[2] The effects of non-metallic inclusions on properties relevant to the performance of steel in structural and mechanical applications http://www.jmrt.com.br/en-the-effects-non-metallic-inclusions-on-articulo-S2238785418304320

[3] Combining EDS-EBSD Analysis to Characterize the Microstructure of a Nitrided Steel https://www.azom.com/article.aspx?ArticleID=17965