Multiscale Modeling – 2nd report

08.01.2019
Marcin Lichota

Abstract

This report applies to the project that was created for a period of 3 laboratory classes. It was created due to the requirements posted on the lecturer (Mateusz Sitko) page. Report covers technology, functionality, examples and summary of achieved results in reference to real world cases.

Technology

As there is a few aspects of technology, I allowed myself to list them below, attaching short description:

- **Application as Web Application** – computer program witch client runs in a web browser, often it requires Web Server to host source files and data. In case of this project, there is created only source code and bundle but there is no Web Server, only in development mode.

- **Programming language JavaScript** – Scripting programming language. Together with HTML and CSS, is a basic tool for building websites. Using the Node.js environment also allows creation of efficient applications operating regardless of the browser. In this project, JavaScript was used as base tool to write application.

- **Git** – distributed version control system for tracking changes in computer files and coordinating work on those files among multiple people. Exactly in the design used for version control and communication with Git Hub repository.

- **React.js** – JavaScript library written by Facebook programmers, used to create graphical user interfaces. It uses declarative, component-based approach. It allows to reuse components across multiple projects and from external sources. In this project React.js is base for whole application. It was used by applying create-react-app that provides boilerplate with react-scripts configurations – best practice configurations for the single page application collected by React Team.

- **Redux –** Predictable state container for JavaScript applications. It helps to write applications that behave consistently based on **Flux** architecture. Redux was used for this project to assure *single source of truth.* It provide limitations that are forcing to one way data flow. Due to that it was much easier to manage current application state.

Functionality

Simple grain growth – taken from the previous project method of grain growth based on the cellular automata technique. In this project it is possible to define up to 600x600 cells surface. It implements 2 neighborhood types (Neumann and Moore) and one transition rule – absorbing boundary condition. This implementation required few factors to make calculations: grid size, cell size and random seed. Random seed is required to initialize seeds that are extended at next steps. As can be seen in the Image 1, functionalities was split into 5 sections described below.

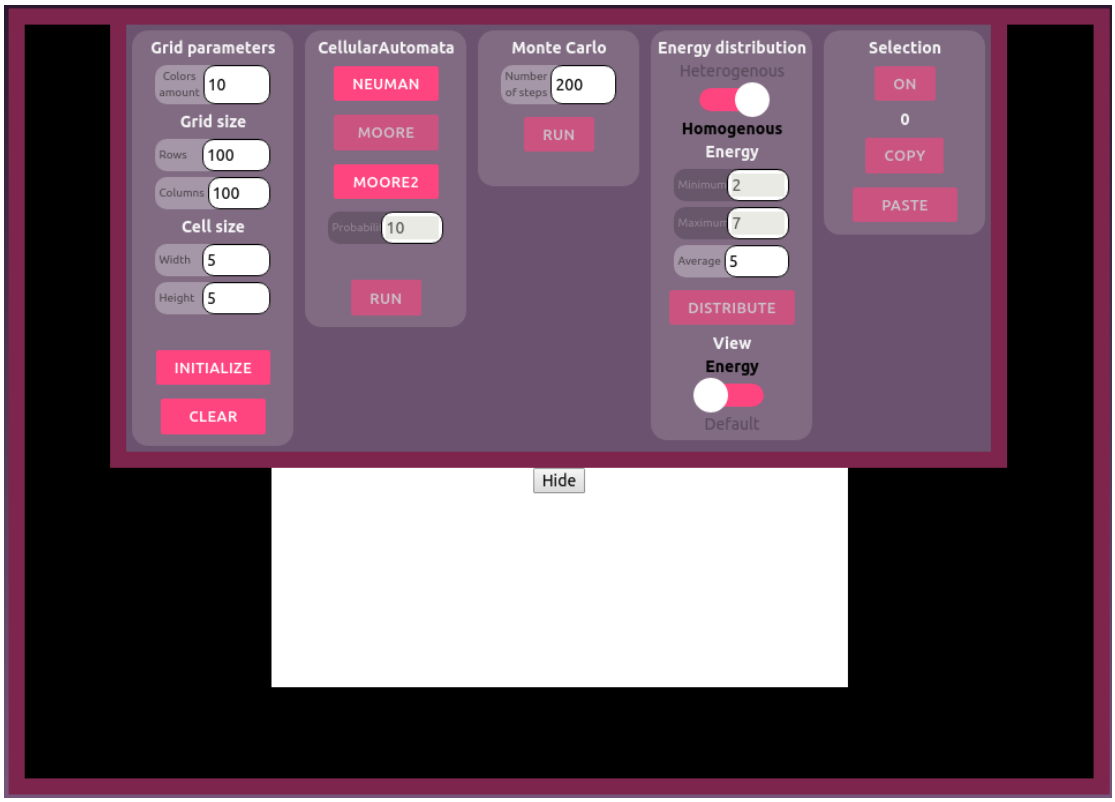

Image 1: Application with expanded menu.

Grid parameters – Necessary to define as this are parameters for the grid description. First parameter (Color amount) is describing how many colors will be used for the next initialization. Initialization is the process of creating new structures and generating new colors in the purpose of further calculations. Next are the grid size and cell size well known from the previous project. Additionally there is Clear button in the purpose of destroying legacy data.

Cellular Automata – This section is also provided from the previous project. It contains 3 methods for the cellular automata grain growth. For the improved Moore method it is required to define probability factor.

Monte Carlo – Totally new section created for the purpose of this project was probabilistic method called Monte Carlo. It's main idea is to generate grains from specified color range for the whole matrix. After that in another step there is consumption of neighboring grains based on the energy calculation and striving a

system to minimize energy. Since in the Monte Carlo method all cells are filled in the initial step. There is different approach to stop procedure. Procedure will not stop when all the cell are filled but rather when iterations number will be equal to the defined number of steps. Images 2, 3, 4 shows example of MC generation for different iterations number.

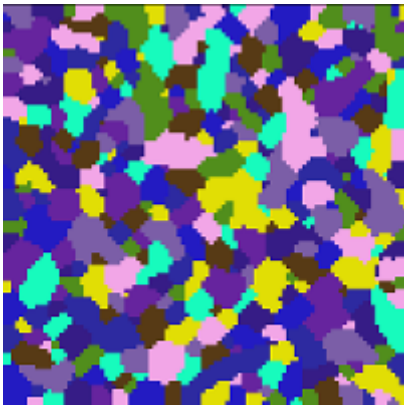

Image 2: Monte Carlo with 100 initial colors and 10 iterations.

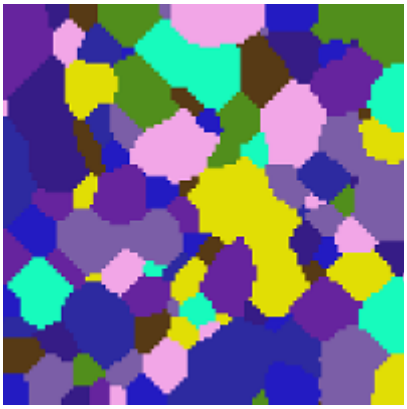

Image 3: Monte Carlo with 100 initial colors and 50 iterations.

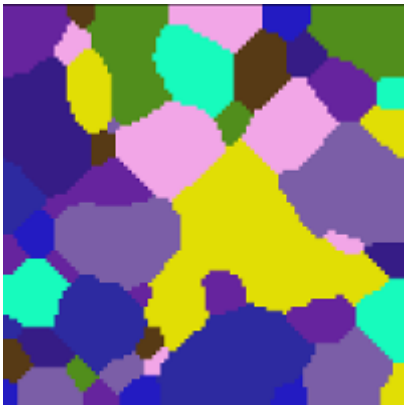

Image 4: Monte Carlo with 100 initial colors and 100 iterations.

Energy distribution – it is a first step of static recrystallization in the Monte Carlo method. In this project two types of the energy distribution is used: homogeneous (“equally” for all cells) and heterogeneous (bigger portion of the energy is stored in the boundary cells, and smaller portion in the rest of cells). In this project selection of the distribution type is possible by the switch button. After selection appropriate type, user should select amount of energy distributed. If homogeneous there is only one input with average energy value and if heterogeneous user should select minimum and maximum value, maximum will be distributed on the edges and minimum into rest of the cells. After this step, user can hit “DISTRIBUTE” button to distribute energy for the currently generated model. After that preview of the energy distribution is applied. User can change it by pressing “View” switch button. Image 5 shows example of energy distribution preview.

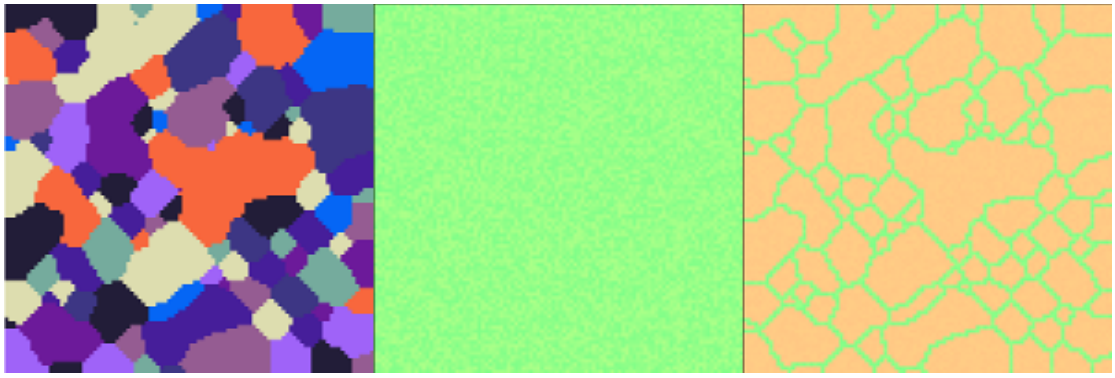

Image 5: Energy distribution: On the left there is orginal model, on the center homogeneous distribution and on the right heterogeneous distribution.

Selection – selection covers feature that allows user to select generated grain and then copy selection in the purpose of the feature paste it in another simulation. User is able to paste selected grains to simulations achieved by the CA and MC methods. This approach lets creates substructures that allows creates dual phase micro structures. Image 6 presents 3 steps of the applying substructure generated by Monte Carlo method to the Cellular Automata - Moore generation.

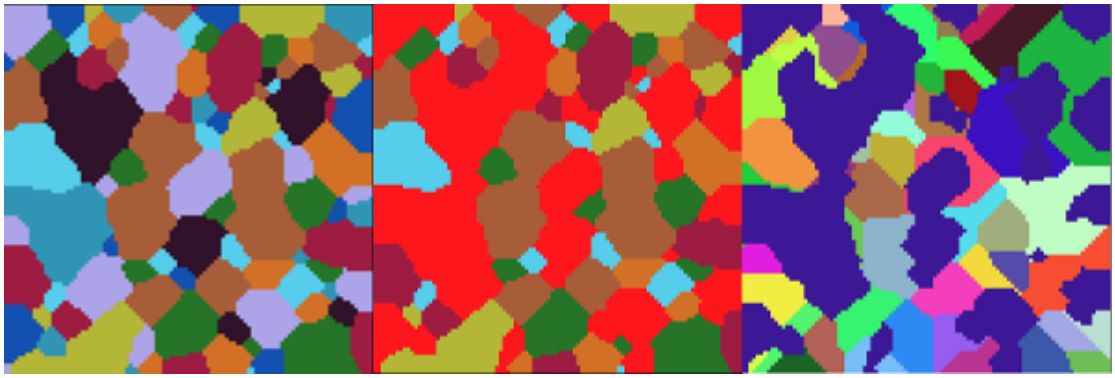

Image 6: Process of applying substructure. In the center picture - selected(red) grains. On the right the same grains pasted to the new model.

Comparison with real cases

Example 1. Cem Tasan, Martin Diehl, Dingshun Yan, Marion Bechtold, Dierk Raabe (2015). An Overview of Dual-Phase Steels: Advances in Microstructure-Oriented Processing and Micromechanically Guided Design.

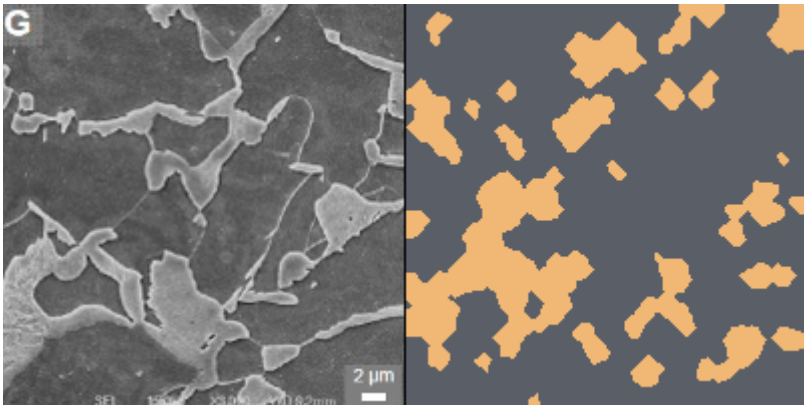

Image 7: On the left: dual phase steel microstructure. On the right: simulation results.

Example 2: Xin Xue, António B. Pereira, JoséAmorim, Juan Liao (2017). Effects of Pulsed Nd:YAG Laser Welding Parameterson Penetration and Microstructure Characterization ofa DP1000 Steel Butt Joint.

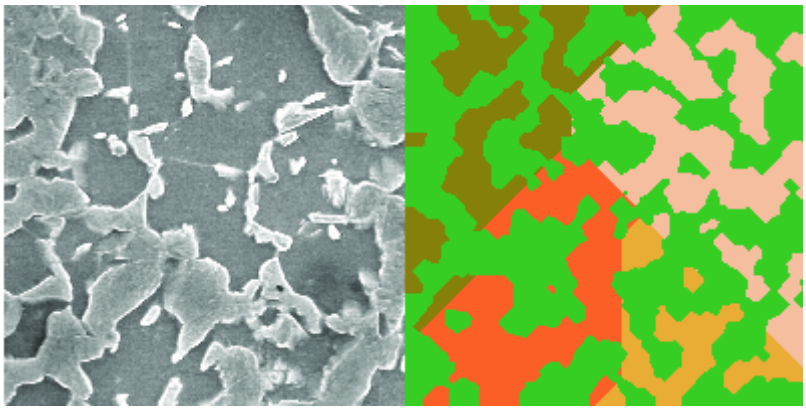

Image 8: On the left: microstructure of dual-phase DP1000 steel. On the right: simulation results.

Summary

The results obtained from the simulation were compared to the actual data. It turned out that feature that allows to gain the best results of accurately reflect the phase structure is grain selection. Generated structures, with some approximation are suitable for further analysis. The program itself has a lot of room to improve its quality both in terms of correctness of the simulation and the user interface.