Corrosion-passivation processes in a cellular automata based simulation study

Łukasz Bartosik, Janusz Stafiej, Dung Di Caprio

Cellular automata model of anodization

Bartosik Łukasz[a], Stafiej Janusz[b], Di Caprio Dung[c]

[a] *Institute of Physical Chemistry, Kasprzaka 44/52, Warsaw, Poland*
[b] *Cardinal Stefan Wyszynski University, Department of Mathemathics and Natural Sciences, Wojcickiego 1/3, Warsaw, Poland*
[c] *PSL Research University, Chimie ParisTech - CNRS, Institut de Recherche de Chimie Paris, 75005, Paris, France*

Abstract

We present a cellular automata model to simulate with a parallelized code nanostructured alumina formation during anodization. The model is based on the Field Assisted Dissolution approach for anodization. The parallel code for model simulation is run on Nvidia Tesla GPU cards. We verify that the parallel algorithm yields correct analytical results for the simple exclusion diffusion between emitting and absorbing walls in 3D. We identify the model parameters that have a strong impact on the nanostructures obtained in simulations and present the diagram of prevalence of these structures. We also simulate in our model the so called "two step anodization" and find an agreement with experimental findings.

Keywords: Anodization, Cellular Automata, Parallel programing, Aluminum oxide growth

1. Introduction

It is known for almost 100 years that several metals can be coated by the passive layer of oxides formed in an electrochemical process called anodization. It has been used extensively to increase corrosion resistance and aesthetical outlook of the surface. Anodization is a rather simple procedure that can be applied

*Corresponding author
Email address: lbartosik@ichf.edu.pl (Bartosik Łukasz)

to a variety of metal surfaces *e. g.* titanium[1], aluminum[2] and hafnium[3]. In the process the surface between the metal and a proper electrolyte solution is polarized by an external voltage producing metal anodic oxidation and related phenomena. The peculiarity of structures that evolve on anodized aluminum surface was first noticed in 1953. Keller, *et. al.*[4] showed with transition electron microscopy that the alumina layer displayed hexagonally ordered pores. For a considerable period of time this phenomenon attracted little attention. A renewed interest comes from its potential nanotechnological applications as scaffolds for other nanostructure synthesis[5]. Masuda *et. al.*[6] determined the experimental procedures for the preparation of regular nanoporous anodic aluminum oxide layers. Then other advancements in the problem started appearing one after another as evidenced by the works of Li[7], Jessensky[8] and Masuda[9]. A major obstacle to obtain nicely organized pore structures is the initial state of the layer. It is fairly disorganized due to defects and initial surface roughness. It needs a thorough pre-treatment of the surface by polishing, chemical and electrochemical etching etc. In this way the so called two step anodization is recommended by Masuda[10]. It consists of two steps. The first step reduces the number of defects and prepares the first network of pores. The layer is then washed out but it leaves an imprint of the pore organization on the metal surface. Then the second anodization produces a better quality organized pore network grown on top of the imprints of the previous one.

The theoretical work on the phenomenon is clearly delayed and unable to describe all the aspects of the process. There are two most considered approaches to base a theoretical description on: Field Assisted Dissolution(FAD)[11] and Field Assisted Flow(FAF)[12]. Sheintuch and Smagina[13] consider the problem of stability for the Field Assisted Flow model by Singh, *et. al*[14] and arrive at the Damped Kuramoto Sivashinsky (DKS) type equation. The model does predict a hexagonally ordered porous layer. But there are two weak points in the approach. The first point is that their analysis applies to the initial stage of the pore growth and cannot guarantee their sustained growth. The second point is that their two dimensional model can be inapplicable in three

dimensions. Indeed, Gomez and Paris[15] explore the regimes of the Damped Kuramoto Sivashinsky equation and conclude that despite stable regimes of the DKS equation in a 2D space they find only chaotic regimes in three dimensions. Stable regimes for the 3D version of the equation need not exist and more research is needed to clarify this issue. In view of the above doubts on viability of 3D FAF model in this work we focus on a 3D Field Assisted Dissolution model.

In this work we design a parallel cellular automata model to simulate the 3D Field Assisted Dissolution model. We explain the rules and states of the model as well as explain our parallelization strategy in the Methods section. We test the strategy on a simple problem of diffusion between emitting and absorbing wall. With the positive test results we pass to simulations of anodization using the verified strategies. We compare our results to theoretical predictions of the Field Assisted Dissolution model

2. Methods

2.1. Cellular automata model

Our model is based on a probabilistic, three dimensional, asynchronous cellular automaton. We consider periodic boundary conditions along the two axes parallel to the initial surface while fixed boundary conditions are applied along the third axis. We use the 3D cubic lattice with the Moore neighborhood for a given cell. In our update scheme each lattice site is given the same chance for an update in one step of system evolution. Then one neighbor is selected from the 3D Moore neighborhood of the updated site. Each neighbor has a uniform probability of 1/26 to be selected as 26 is the coordination number for the Moore neighborhood in 3D. Let us note that the pair of sites thus selected has a double chance for selection since the neighbor and the site in another instance may change the roles. The pair thus selected is subject to an update and the rules of updates are symmetrical in that they ignore the distinction between the site and the neighbor. The pair as a whole is updated according to the formal two body reaction scheme depending on the states of the two sites.

We assume that 5 cell states describe adequately the physicochemical content of our model. These are: M the state corresponding to the metal, OX - to the oxide, EF - to the walker representation for electric field in the oxide that will be detailed further, A - to anion rich sites in the oxide S - to solvent. Let us note that the modeled oxide layer is represented by cells in one of the three states: OX, EF and A. We call them *oxide-like* states. All of these cells are considered to be filled with oxide and two of them EF and A have an additional function. States EF should be interpreted as OX electrostatic field carriers such that the average flux of these particles in a stationary state yields the magnitude of electric field. States A are considered as anion (typically superoxide O^{2-} or hydroxide OH^{-} ions) rich OX oxide and the anions are dragged by the electric field towards the metal anode. It is convenient often to think of states as of particles. Our update rules apply to two particles at two neighboring sites A and B where at update they can undergo a binary reaction:

$$Particle1(position\ A) + Particle2(position\ B) = Particle3(position\ A) + Particle4(position\ B)$$

which can be selected with a given probability relative to the obvious alternative reaction channel, in which nothing is changed and leaving the selected pair of sites in their actual state:

$$Particle1(position\ A) + Particle2(position\ B) = Particle1(position\ A) + Particle2(position\ B)$$

For example the following reaction

$$M + OX = M + EF \tag{1}$$

should be therefore read as: An M particle in site A reacts with an OX particle in site B yielding an M particle in site A and an EF particle in site B. In the above reaction, we omit an explicit notation for the sites A and B, as in the course of the simulation they correspond to a random pair of neighbouring sites chosen with the algorithm detailed thereafter.

Let us note that the order in the equation is relevant and in our convention the particle at A comes before that at B:

$$M + OX = M + EF \tag{2}$$

has a different final effect than the reaction:

$$M + OX = EF + M \tag{3}$$

The difference comes from the fact that in reaction 2 the M particle stays at the site A while in reaction 3 it moves from site A to site B. As mentioned above neighbouring sites A,B are equivalent to B,A. Thus reaction

$$OX + M = EF + M \tag{4}$$

is equivalent to reaction 2 and need not be duplicated.

The rules we postulate are as follows:

$$M + S = OX + OX \tag{5}$$

$$EF + S = S + S \tag{6}$$

$$EF + S = A + S \tag{7}$$

$$M + A = OX + OX \tag{8}$$

$$M + OX = M + EF \tag{9}$$

$$EF + OX = OX + EF \tag{10}$$

$$EF + A = A + EF \tag{11}$$

$$S + OX = OX + S \tag{12}$$

$$S + A = A + S \tag{13}$$

These 9 rules can be divided into three groups: reaction-like rules (5 to 9), diffusion-like rules (10 and 11) and surface reorganization (12 and 13). Let us rephrase you mean emphasize ? that states OX, EF and A are all *oxide-like* states, containing a portion of oxide. In the following paragraphs we explain how we apply the rules in actual coding. Four of the five reaction-like rules can be viewed as descriptions of the well known electrochemical processes taking place during anodization. These are:

1. Passivation of the active metal (rule 5)
2. Dissolution of the oxide in high electric field (rule 6)
3. Dissociation of the solvent and anion incorporation into the oxide layer in high electric field (rule 7)
4. Oxidation of the active metal by the incorporated anions (rule 8)

The final reaction-like rule creates electric field walkers that we use to model the presence of electric field in the oxide layer. A detailed explanation of how we account for the electric field presence is given in the next subsection. All of the reaction-like rules have a fixed probability during simulations. Additionally the probabilities of rules 6 and 7 sum to 1.

The diffusion-like rules govern the Random Walk of electric field walkers and the movement of anions inside the oxide layer. The Random Walk is modeled by two appropriate sites swapping their positions. These rules differ from reaction-like rules in the sense that they conserve the species involved leading to their spatial redistribution. For the most part we set their probability to 1 if the correct particles encounter one another. However in all other aspects they are identical to reaction-like rules and are parallelized in the same way. In our model we assume that anion movement is mostly caused by the strong electric field present in the oxide layer. During anodization the electric field arises from the onset of the voltage between the metal and the solution. The electric field is strongest at the thinnest metal-solution separation by the oxide layer and becomes weaker as the oxide layers gets thicker. Rule 11 represents this behavior well. It creates the drift of anions in the direction opposite to the gradient of the electric field walker concentration. The concentration gradient of the electric field walker is balanced in quasistationary conditions by the electric walker flux and then both can be thought to mimic the presence of electric field.

Rules 12 and 13 govern surface reorganization. The goal is to introduce a form of surface tension to prevent excessive ramification of the obtained structures. Oxide particles are held in place by bonding to other oxide particles. Let us recall that *oxide-like* means *oxide*, *anion* and *electric field* states as they all

are present within the oxide layer. These rules remind diffusion but there is a difference with the diffusion rules mentioned previously. The decision if a swap of particles occurs is based on two factors: the neighborhoods of the locations chosen for the swap and a predefined probability, P_{BOND}, called "unbonding" probability. If the *solvent* site has more oxide neighbors then the *oxide* or *anion* site a swap occurs. In the opposite case a swap may still occur with a probability given by the power law:

$$P = P_{BOND}^{N} \tag{14}$$

where P_{BOND} is the "unbonding" probability and N is the difference of the number of neighbors between the sites. Additionally if an oxide site has no oxide neighbors it dissolves into a solvent. The relative frequency of reaction-like, diffusion-like and surface reorganization events can be set at the beginning of the simulations.

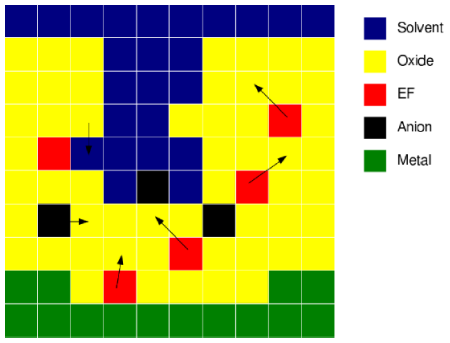

Figure 1: A schematic 2D representation of a pore forming in the model. Arrows connect a chosen site to its selected nearest neighbor. States are depicted by different colors.

2.2. Modeling the effect of electric field

The aspect of this work worth pointing out is the way the presence of electric field is accounted for. We share many general features with the model of anodization developed by Parkhutik and Shershulsky[16]. To keep it simple we treat the metal-oxide and oxide solvent surfaces as equipotential and thus avoid hypotheses about eventual double layers that appear in a microscopic description for metal-layer and layer-solvent interface:

$$\varphi_{Ox\backslash S} = 0 \tag{15}$$

$$\varphi_{M\backslash Ox} = U_{M\backslash Ox} \tag{16}$$

where $\varphi_{Ox\backslash S}$ is the potential at the oxide-solvent interface, $\varphi_{M\backslash Ox}$ is the potential at the metal-oxide interface and $U_{M\backslash Ox}$ is the voltage applied to the system. To find the electric field we use the Poisson equation:

$$\nabla^2\varphi = \frac{\rho_f}{\epsilon} \tag{17}$$

where ρ_f is the free charges density and ϵ is the electric permittivity of the layer. We neglect the free charges contribution so $\rho_f = 0$ and the Poisson equation is reduced to a Laplace equation

$$\nabla^2\varphi = 0 \tag{18}$$

The Laplace equation is also encountered in a problem of diffusion defined as:

$$c_{interface1} = 0 \tag{19}$$

$$c_{interface2} = c_0 \tag{20}$$

where $c_{interfacen}$ is the concentration of diffusing particles at particular interfaces and c_0 is a fixed concentration. The Fick's diffusion equation is as follows:

$$\frac{\partial c}{\partial t} = D\nabla^2 c \tag{21}$$

In steady state $\dfrac{\partial c}{\partial t} = 0$ and the equation simplifies to a Laplace equation:

$$\nabla^2 c = 0 \tag{22}$$

We therefore conclude that one can solve the Poisson equation in this case via methods used to solve the steady-state diffusion equation. It is well established that the steady-state diffusion equation can be solved using the Random Walk [17]. In the case of anodization no steady-state is reached, and both the metal/oxide and oxide/solvent interface move with time. We assume that a quasi-stationary state is present in the system and hence $\frac{\partial c}{\partial t} \approx 0$. We justify our assumption on the basis that the rate of diffusion is at least two orders of magnitude greater than the rate of interface movement. This stems from the fact that one diffusion step is taken per time step. And typically for the parameters used in our simulation, the interfaces move overall approximately 150 lattice sites in about 40000 time-steps. With other parameters our assumption need not hold as the walker distribution may not reach the stationary form. With these stipulations in mind we decide to employ the Random Walk approach to model the electric field in our system via the diffusion of *EF* walkers. We can then follow the rearrangement of the electric field distribution which conditions the reactions at the interfaces in terms of their spatial evolution.

2.3. Parallelization

In order to shorten the time required for the simulations we use of the capabilities of GPU. Employing CUDA Tesla cards requires however running a parallelized program. The abbreviation CUDA stands for Compute Unified Device Architecture [18]. We base our approach to parallelization on the block-synchronous method presented by Olga Bandman [**?** **?**]. The updates according to each rule-type group should need to occur with uniform probabilities for every site in the lattice. Let us recall that a kernel is a CPU launched procedure executed on GPU controlling multiple threads running in parallel bunches. Thus we organize our parallel computing in three separate kernels: reaction-like rule kernel, diffusion-like rule kernel and surface reorganization kernel. In our mind the kernel launch of one of the three types constitutes a substep in the single time step of the simulation. The number of substeps to make a step comes from the following organization of the calculation. We divide the grid of sites

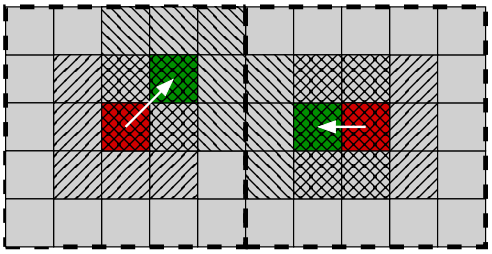

Figure 2: The parallelization scheme for two adjacent 5×5 blocks on a 2D example. The red and green squares represent respectively the active site and its selected neighbor with their respective neighborhoods hatched right and left and in both directions for common sites. The 5×5 blocks guarantee that no overlap takes place between neighborhoods from two adjacent blocks.

of our simulation box into a cubic lattice with the cube size 5 by 5 by 5 sites and decide that the simulation box sizes must be multiples of 5. The origin for the coordinate system of this lattice is selected at random from among all the sites of the box at each substep. Because of the lattice periodicity in the three principal directions it amounts to selecting 125 origin points at random in the base cube positioned at (0,0,0) site of the simulation box coordinate system. The kernel launched at the substep updates simultaneously all the sites in the centers the cubes contained within the simulation box with due regard for periodic conditions for cubes on the boundaries of the box. Thus during 125 substeps each point gets a chance to be treated once on the average by a specific kernel that makes 375 substeps for the three kernel types. They are launched in a randomized order to avoid spurious correlations. The size 5 is selected to avoid update collisions from different threads working in parallel on updated points. In principle 3 is enough for reaction-like and diffusion-like kernels as they involve only Moore neighbors of the updated site. However the surface-diffusion kernel updates the site counting the neighbors of the neighbor and we extended the size to 5 keeping it the same for all three cases.

Such partitioning of the grid allows us to use the so called shared memory of the Tesla cards to improve the calculation efficiency compared to using global card memory. Shared memory is unique to a given block of threads. Block

of threads is a concept pertinent to the CUDA card architecture. Threads organized in one block can access the shared per block memory. Threads from different blocks can only interact via the global memory. The shared memory is much faster in performance than the global memory but limited in size. At each substep, from global memory, the relevant states in the grid are copied into the shared memory of each block for threads to process. This allows threads for a faster access to the information on the neighborhoods of updated sites and it is particularly efficient in the case of surface diffusion (data not shown).? After a given substep the global memory of the card is updated and another substep commences. We have verified that in some cases the code using shared memory is 5 times faster than that based on global memory updates only. With a predefined data collection frequency the content of global GPU memory is copied from the device onto the CPU and output to files. It is mainly in the CPU that simulation data are processed to extract the shapes of the interfaces, Fourier transforms of cross-sections, distributions of particles. This can be done with conventional tools so that CUDA resources are not engaged in these tasks.

3. Results and discussion

All simulations are performed (to avoid using twice results/results) on the Tesla C2070 card. The initial results of parallelization yield a significant increase in simulation speed compared to sequential code and we decided to work exclusively with the parallel code. A comparative study of the code efficiency certainly interesting in other context is beyond our scope.

3.1. Diffusion

First we verify if our approach to diffusion and parallelization yields reasonable physical results. We simulate a system composed of a single type of particles in a three dimensional lattice. In this system the bottom layer sites are filled with diffusing particles. The particles disappear when they arrive / arriving at the top layer on the opposite side of the box. The analytic solution of the problem of steady state diffusion in this geometry is well known. According to it the particles form a linear distribution along the direction perpendicular to the sides. We test for any possible effect of spurious correlations that may arise in our scheme of update in the parallel algorithm. The linearity of the distribution of the diffusing particles is the indication that our system is free from such effects. To confirm our findings we calculate the diffusion coefficient resulting from our simulations and compare it to the theoretical diffusion coefficient that we should obtain for a three dimensional cellular automaton using the Moore neighborhood. The value of the theoretical diffusion coefficient is obtained using the following calculation:

$$< R^2 >= 2D \times d \times T \tag{23}$$

where $< R^2 >$ is the mean square displacement, D is the homogeneous diffusion coefficient, T is time and d is the dimensionality. In our case $d = 3$.
The mean square displacement calculated for the Moore neighborhood is given as follows. There are a total of 26 neighbors to a site, 6 of them are at square distance 1, these points are the centers of the cubes walls, 12 are at a square

distance 2 each of them at the center of cube edges and the corners of the cubes (8 points) are at a square distance 3. The mean square displacement is therefore: $\frac{6 \times 1 + 12 \times 2 + 8 \times 3}{26} = \frac{54}{26}$. Taking a single time step (T=1) we obtain the equation:

$$54/26 = 6D \tag{24}$$

$$D = 0.346154 \tag{25}$$

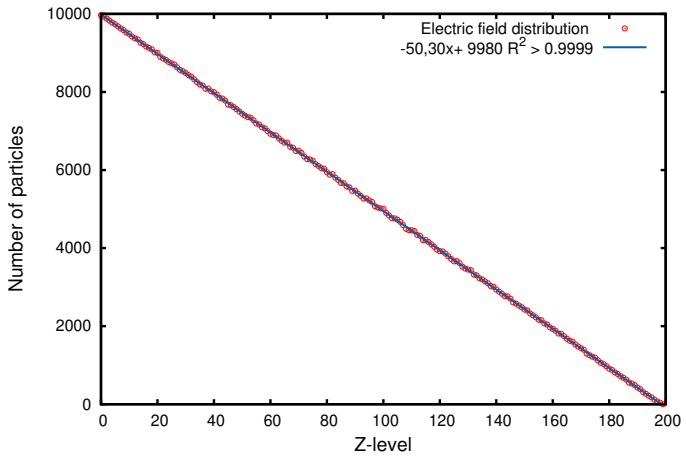

Figure 3: Number of particles distribution profile.

We compare this theoretical value to values obtained in our simulation via the Fick equation:

$$J = -D \times \frac{\Delta c}{\Delta x} \tag{26}$$

The flux J is obtained by counting the particle creation and disappearance events at opposite boundaries of the system along the Z axis. We assume that the steady state is reached when number of annihilations approaches the number of creations and both are equal within the relatively small fluctuation. Then we average the distribution for a period of simulation time steps. This average particle distribution over the simulation time period after reaching the steady state is plotted and a linear fit of the result is performed via standard methods. As seen in figure 3 the R^2 parameter for the linear regression is over 0.9999

which signifies that the distribution plot is a straight line indeed. Otherwise we would be forced to admit that the spurious effects prevent the program from reproducing well established diffusion behavior. The average numbers of particles created and destroyed in a specific time interval gives us an independent estimate of the diffusion coefficient. It is readily obtained from the common value of these numbers divided by the absolute value of the slope of the curve and additionally divided by 2 due to the symmetry of the used procedures with every pair of sites having two chances to undergo a diffusion step. Our simulation yields the diffusion coefficient $D = 0.34602$, in the range of 0.1% from the expected result of 0.346154.

The simplest Random Walk numerical experiments consider a single particle displaced at random to neighboring sites. Our parallel approach converges with these experiments well in the case of low concentrations of moving particles. Then the probability of two mobile species meeting and one interfering with another's movement is low. The movement of each particle may be considered independent. This assumption need not hold in regions of high concentrations of moving particles and one may not rule out an effect on the diffusion process. However, lack of deviation from linearity even in regions of high concentrations (close to 1 particle per site) is a good indication that if such an effect exist it is negligible. We conclude that our procedure to simulate diffusion does not contain any significant error leading to obviously unrealistic results and can be considered for steady-state diffusion even at high concentrations.

3.2. Ordered structures

In this section we present the results of simulations using our approach. All of the simulations were conducted in systems of 100 by 100 by 200 sites using constant simulation parameters throughout each run. Regardless of the parameters all simulations concluded within 30000 to 35000 time steps. Figure 4 contains a side view of the oxide-solvent interface in one of our systems. A porous structure is clearly visible, however not much can be said about how organized it is.

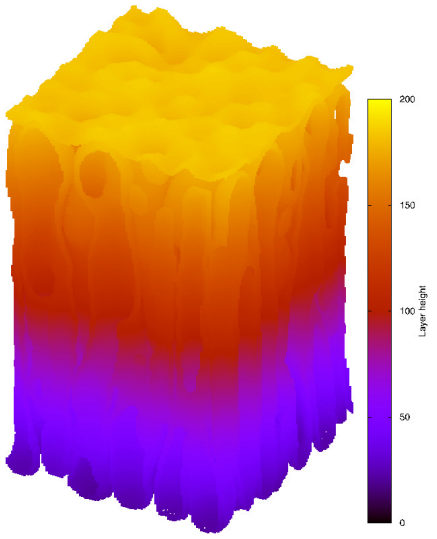

Figure 4: A side view of a simulated oxide layer. $Ox \backslash S$ interface is shown.

300 To address this problem we present a cross-section taken at half oxide layer height and a Fourier transform of this cross-section in figure 5. The cross-section confirms the existence of developed well separated pores. The Fourier transform indicates a hexagonal symmetry. It is noteworthy to acknowledge the neighborhood selected for the cellular automaton is the Moore neighborhood, which has a quadratic symmetry. We thus conclude that the observed hexagonal pattern does not originate from our choice of the neighborhood for the automaton. We further calculate the Fourier transform modulus with respect to the wave vector modulus to find a characteristic length between the pores. The results are presented in figure 6. The peak in figure 6 corresponds to the characteristic length present in the system. In figure 7 we present the distribution of *oxide-like* states (*oxide*, *electric field* and *anion*) along the Z-axis of the system. This distribution can be described as having two peaks and a plateau between them. The peak corresponding to low Z-axis values corresponds to pore bottoms, where more oxide material is present. The plateau is the region where a stable porous layer exists. The second peak nearing the "top" of the system is the result of the simulations starting conditions. Starting the simulation from a planar

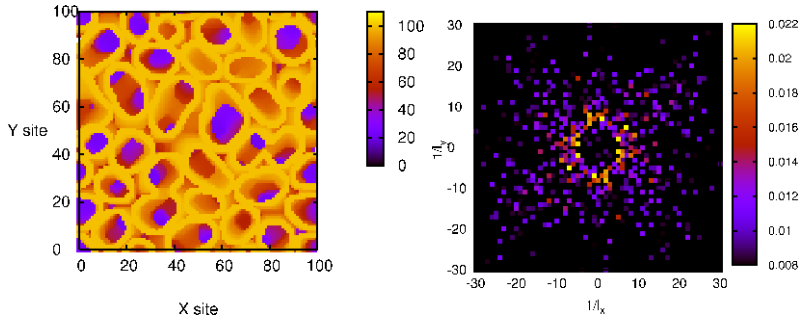

Figure 5: A cross-section of a organized porous layer (left) and the Fourier transform of the cross-section(right). In yellow the $Ox\backslash S$ interface, the oxide in between pores is not shown.

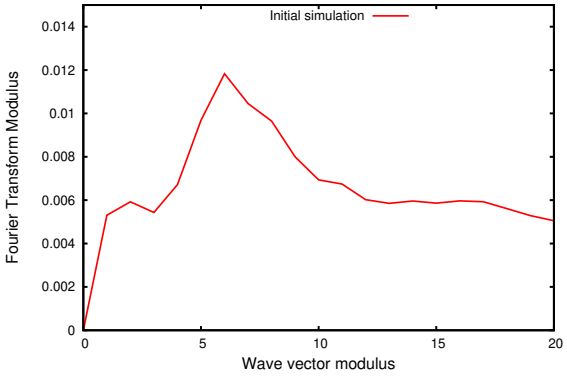

Figure 6: Wave vector modulus in the cross-section.

geometry between the metal and solvent layers creates an overflow of oxide material at the initial stage of the simulation (passivation of the active metal rule). This material then aggregates due to surface tension and stays relatively stable throughout the simulation.

3.3. Two step anodization

In order to test our simulation against a real world experiment we decide to simulate two step anodization. The expected results is a better organized porous structure with an unchanged characteristic length and shape. We use a metal-oxide interface obtained in a source simulation as the starting interface for the second step of anodization. This is meant to model the washing off

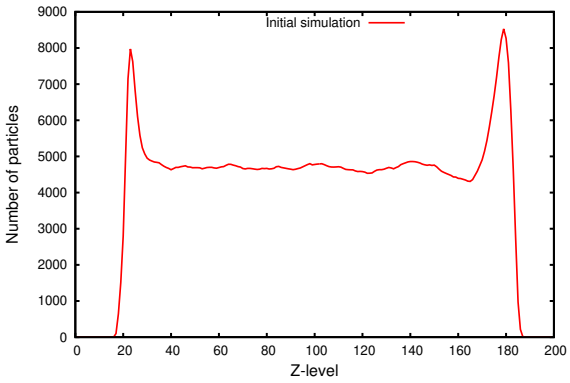

Figure 7: Oxide layer profile

the oxide layer between anodization steps in experimental procedures. Aside of the initial conditions all simulation parameters were kept constant in both simulations. A porous structure is clearly visible on cross-sections of the layers. In figure 8, we present a side by side comparison of simulations of single step and two step anodizations.

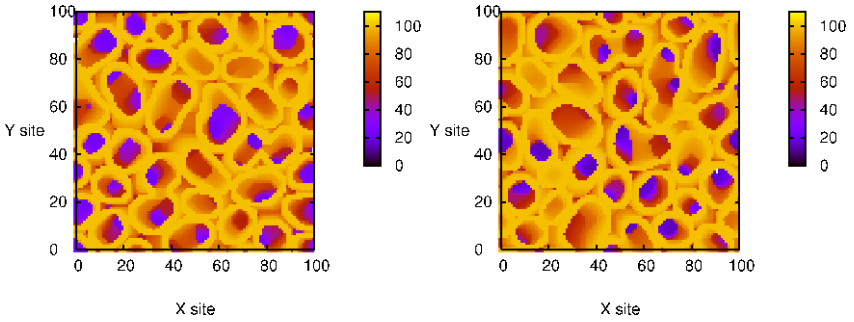

Figure 8: Comparison of cross-section of source (left) and loaded (right) simulation.

Both cross-sections are very similar between the simulations and both show similar porous structure. It is impossible to tell if an improvement in organization has been achieved just from these results. To highlight the differences between the layers we compare the Fourier transforms of the above cross sections in figure 9 and the comparison of Fourier transform moduli in figure 10 .

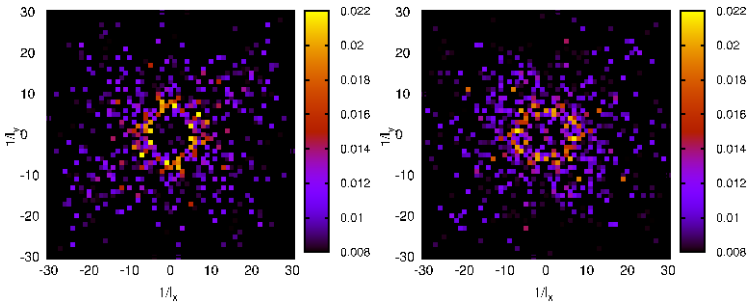

Figure 9: Comparison of Fourier transforms of cross-sections of source (left) and loaded (right) simulation.

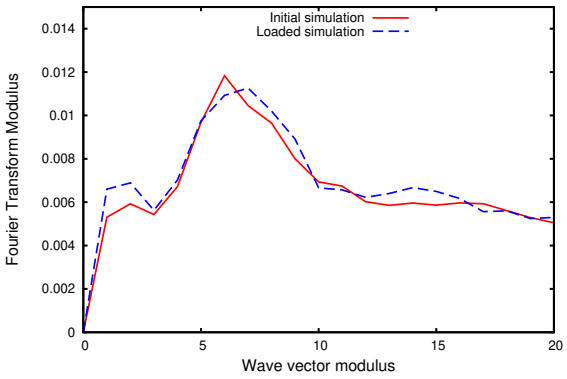

Figure 10: Fourier transform modulus comparison.

The Fourier transforms show that the characteristic length of the system has not changed between the simulations. However, a hexagonal pattern is much more apparent in the two step simulation transform. Figure 10 shows that the location of the peak in Fourier transform modulus is unchanged, meaning that the characteristic length of the system is the same in both simulations. These result signify that the layer obtained from a previously obtained interface is indeed more organized compared to the original layer. Last we compare the particle profiles in figure 11. The overall shape of the profiles is similar, however two differences require an explanation. The first difference is the slight shift of the first peak (corresponding to pore bottoms) towards lower Z-axis value. The second is different shape of the second peak near the "top" of the system. Both of

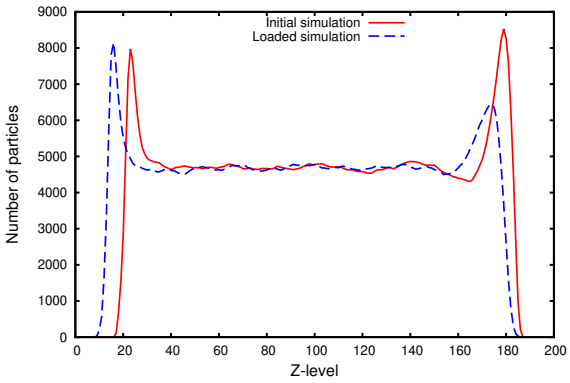

Figure 11: Particle profile comparison

these changes can be attributed to different starting conditions of the initial and loaded simulation. During the initial simulation the starting layer was planar, while in the loaded simulation pore bottoms of the depth of approximately 8-12 were embedded into the structure. This fact means that some particles of solvent of the loaded system were initially lower along the Z-axis than analogous particles in the initial system. The change of the shape of the "top" peak can be explained much in the same manner. The early stage oxide structures were different in both cases producing a different outcome globally. Despite these differences the overall shape of the distribution is similar in both simulations and the concentrations of *oxide-like* particles along the Z-axis remains unchanged between the simulations. This corresponds to both structures displaying the same porosity.

3.4. Structure diagram

We present our data from one hundred simulations in figure 12. The aim of these simulations was to investigate the influence of two parameters: "unbonding" probability and oxide dissolution chance in high electric field. Simulations show a variety of results. These result are in good agreement with theoretical predictions of Stanton and Golovin [19]. There is a clearly visible region of organized porous structures surrounded by regions of unorganized pores and

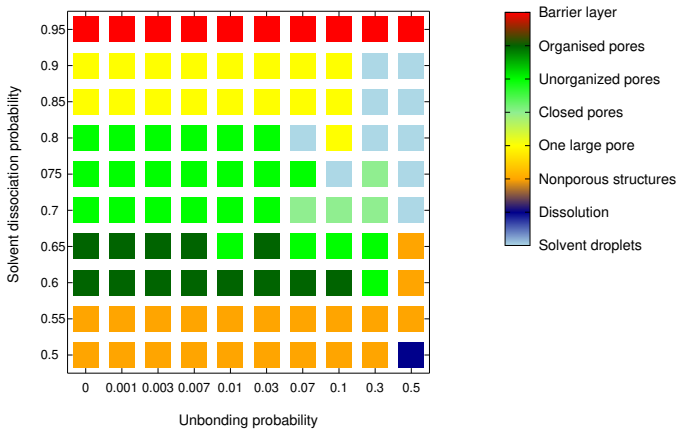

Figure 12: Structure diagram.

non-porous structures. The impact of both investigated parameters is visible, "unbonding" probability seems to have a particularly significant effect. In figure 13, we present some non-porous structure which we have obtained. The existence of such structures highlights the possibility of using our model or its derivative to simulate organized structures other than porous layers.

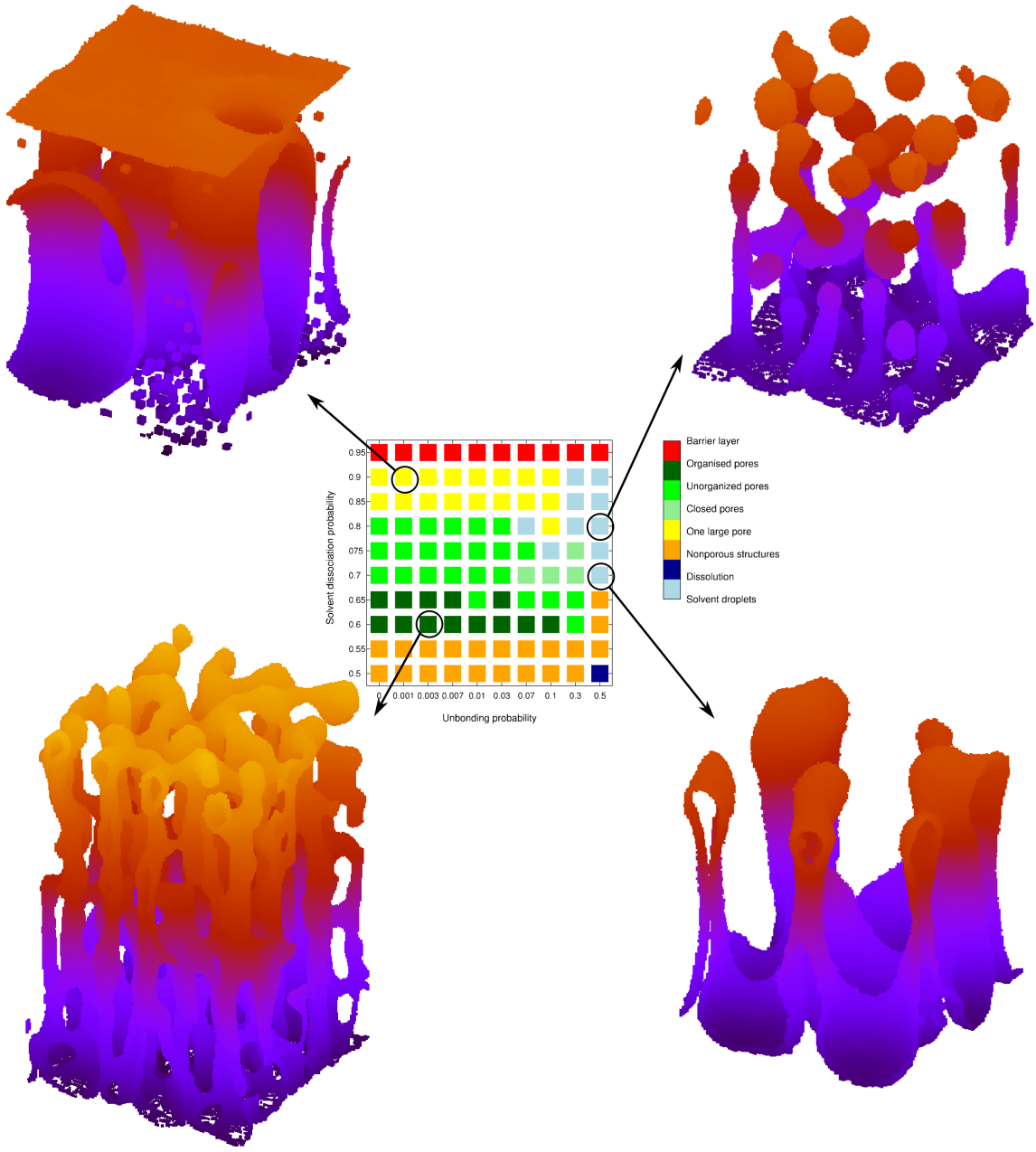

Figure 13: Examples of simulated structures. $Ox\backslash S$ interface is shown.

4. Conclusions

We have designed and implemented a three dimensional asynchronous cellular automaton for the purpose of simulating anodization of aluminum. To that end we employed the computational power of GPU on Tesla Nvidia cards. We introduce to the parallelization scheme features related to CUDA resources such as on board random number generation and fast access shared block memory. The algorithm is tested on a simple three dimensional diffusion problem. Results of this test are consistent with predictions and we believe that our algorithm does not cause unphysical behavior to emerge. Results of our simulations match other theoretical models on a qualitative level and we were able to obtain an ordered hexagonally symmetric porous layer. Moreover we have reproduced the qualitative effect of performing two step anodization which results in an improvement in ordering of the structure if the anodization is performed on a pre-treated metal surface.

5. Acknowledgements

The work of Ł. B. was realized within the International PhD Projects Programme of the Foundation for Polish Science, cofinanced from European Regional Development Fund within Innovative Economy Operational Programme "Grants for innovation" . The work has also benefited from Polish NCN grant no. N N204 139038.

6. References

[1] J. Wang, L. Zhao, V. Y.-S. Lyn, Z. Lin, Journal of Material Chemistry 19 (2009) 3682–3687.

[2] H. Rezazadeh, M. Ebrahimzadeh, M. R. Z. Yam, World Academy of Science, Engineering and Technology 70 (2012) 30–32.

[3] S. J. Garcia-Vergara, et al., Electrochimica Acta 54 (2009) 3662–3670.

[4] F. Keller, M. Hunter, D. Robinson, Journal of the Electrochemical Society 100 (1953) 411–419.

[5] J. A. Mutalib, D. Losic, N. Voelcker, Progress. Mater. Sci. 58 (2013) 636–704.

[6] H. Masuda, K. Fukuda, Science 268 (1995) 1466–1468.

[7] A. P. Li, et al., Journal of Applied Physics 84 (1998) 6023–6026.

[8] O. Jessensky, F. Muller, U. Gosele, Journal of the Electrochemical Society 145 (1998) 3735–3740.

[9] H. Masuda, K. Yada, A. Osaka, Japanese Journal of Applied Physics Part 2 - Letters 37 (1998) L1340–L1342.

[10] H. Masuda, M. Satoh, Japanese Journal of Applied physics Part 2 - Letters 35 (1998) L126–L129.

[11] T. P. Hoar, N. F. Mott, Journal of Physics and Chemistry of Solids 9 (1959) 97–99.

[12] S. Garcia-Vergara, L. Iglesias-Rubianes, C. Blanco-Pinzon, P. Skeldon, G. Thompson, P. Campestrini, Proc. R. Soc. A 462 (2006) 2345–2358.

[13] M. Sheintuch, Y. Smagina, Physica D 226 (2007) 95–105.

[14] G. K. Singh, A. A. Golovin, I. S. Aranson, et al., Europhys. Lett. 70 (2005) 836–842.

[15] H. Gomez, J. Paris, Physical Review E 83 (2011) 046702–1;046702–11.

[16] V. P. Parkhutik, V. I. Shershulsky, J. Phys. D.: Appl. Phys. 25 (1992) 1258–1263.

[17] R. Gorenflo, F. Mainardi, Fractional Calculus & Applied Analysis 1 (1998) 167–191.

[18] `http://www.nvidia.com/object/cuda_home_new.html`.

[19] L. Stanton, A. A. Golovin, Physical Review B 79 (2009) 035414–1–035414–7.