Project Final Report

Multicrystalline Growth Simulation

Link to Git repository:
https://github.gatech.edu/zliu629/CSE6730-vanderDriftSimulation

Team 36

Zhaonan Liu
Shikai Jin
Yifan Ma

A project final report for the class of
CSE 6730
Georgia Insitute of Technology

1 Abstract

Zinc is a promising anode material for rechargeable batteries because of its high energy density, low cost, and environmental friendliness. Within the electrochemical cell, electrodeposition of zinc is a neccessary process that deposit zinc metal on a conductive substrate by applying an electric current. Currently, electrodeposition of zinc also poses some challenges, such as uncontrolled morphology, dendrite growth, and interfacial side reactions. In terms of morphology, (002)-textured zinc electrodes are highly desirable since they have excellent reversibility and stability in aqueous electrolytes. Recently, it was reported that the morphology of zinc electrode may be correlated with the rate of electrodeposition by varying the applied current density in the electrochemical cell. However, the mechanism of polycrystalline growth that yield the observed difference in orientation preference and morphology remains unclear. In this study, we investigate the polycrystalline growth in electrodeposition by simulating the growth of polycrystal from random nuclei. Ultimately, the objective is to correlate the surface morphology obtained from simulation with those observed in experiments and verify the influences of various parameters on the evolution of texture. With the complex nature of this simulation and limited time, we were able to simulate and visualize nucleation and locallized polycrystalline growth. To validate our simulation, we inspected different types of interaction between grains in the simulation and compare them to scanning electron microscopy (SEM) images.

2 Project description

The model system we currently consider is an idealized electrode surface during electrodeposition. During electrodeposition process, metal ions in the electrolyte get reduced on the working substrate and form nuclei and keep growing. The purpose of this simulation is to investigate how Zn crystal grows as a function of time during electrodeposition process. In our simulation, the system is a 3D space with a stationary substrate containing predetermined nucleation sites where each crystal may grow with hexagonal closest packed (hcp) structure in x, y and +z direction until it makes contact with other surfaces. The total polycrystalline growth rate within the space is fixed (corresponding to the current density) and the growth rate of individual surfaces is determined according to the discussion in section 2.4. The objective of this project is to simulate the growth process of polycrystal and compare the resulting surface morphology with experimental findings.

3 Literature review

Electrodeposition of zinc is a process of depositing zinc metal on a conductive substrate by applying an electric current. A typical electrodeposition setup is shown in Figure 1. Zinc is a promising anode material for rechargeable batteries because of its high energy density, low cost, and environmental friendliness. However, electrodeposition of zinc also poses some challenges, such as uncontrolled morphology, dendrite growth, and interfacial side reactions, which can affect the cycling stability and performance of the batteries. Therefore, various strategies have been proposed to regulate the electrodeposition behavior of zinc, such as adding surfactants, using separators, modifying the current density, and employing 3D substrates. These strategies aim to improve the reversibility, uniformity, and structure of the zinc deposits, and to enhance the electrochemical properties of the zinc-based batteries.

(002)-textured zinc electrodes are highly desirable since they have excellent reversibility and stability in aqueous electrolytes. A method to produce (002)-textured zinc electrodes is reported in recent literature. It has been found out that the texture of electrodeposited zinc can be controlled by modifying the current density. Higher current density tends to result in a dendrite-free (002)-textured zinc electrode. Two articles on this topic have given different explanations why a (002)-textured zinc electrode is derived. The first article explained this phenomenon by the law of Bravais [2]. The fast-growing facets are the first to disappear, while the slower-growing facets are exposed more. The authors stated that higher current density provides extra energy to boost the growth rate of the (100) crystal plane, and thus exposes more (002) crystal planes. However, this explanation is tangential to the phenomenon. A (002)-textured zinc electrode means that the

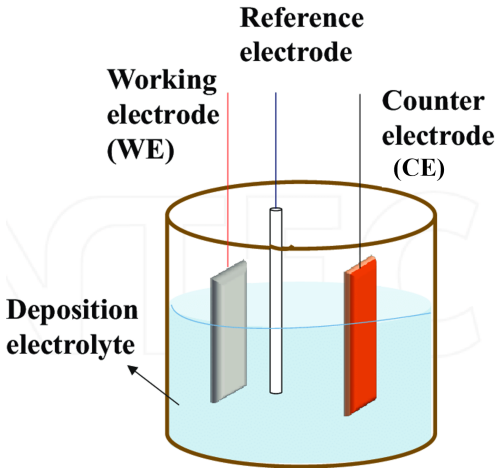


Figure 1: A typical 3-electrode electrodeposition setup.[1]

crystal grains with the orientation where (002) crystal planes are parallel to the substrate are dominant, not that more (002) surfaces are exposed. The second article attributed the result to the effect of hydrogen evolution on the nucleation of zinc [3]. However, according to one of the teammates in our group, in-situ X-ray diffraction (XRD) data has shown that the nucleation is largely random, and the texture is formed during the electrodeposition process. Therefore, the explanation in the second article is also incorrect.

Based on our observations, we decided to investigate polycrystalline growth in electrodeposition with simulation. The first type of crystal growth simulation is to model the growth of single crystals, typically done using molecular dynamics (MD) and Monte Carlo (MC) methods. MD simulations involve modeling the motion of atoms and molecules in a crystal lattice. However, the rate of crystal growth processes is typically much slower compared to the atomic/molecular motion, making most direct MD simulation of crystal growth impractical. On the other hand, MC simulation involves random sampling of crystal growth processes to estimate the probability of crystal formation and is often in time the more feasible approach [4]. MC simulations of growth are based on simplified growth models with approximated geometrical and dynamic rules. Some of the recent works included a general approach for predicting crystal growth proposed by Anderson and colleagues, which uses a unified kinetic MC with 3D partition to predict crystal habit and nanoscopic surface topography simultaneously [5]. Based on this idea, CrystalGrower, a MC crystal growth simulation tool, is developed [6]. In addition, cellular automata is also an effective method for gnerating dynamic topological models of crystal structure. It has shown remarkable success in modeling ice crystal growth, and was the first to produce realistic snow-crystal structures that are both faceted and dendritic in 3D. [7].

The second type of crystal growth simulation is on a more macroscopic scale. This type of crystal growth simulation treats polycrystalline grains as geometric shapes, or polyhedra. An important motivation behind this simulation is evolutionary selection, a theory brought up by van der Drift in 1967 [8]. As the polycrystalline thin film grows, the grains compete with each other and some will be completely covered by other grains and cease growing. Simulation results have verified that this process indeed selects growing grains in certain orientations, resulting in a textured surface in the end. The in-situ XRD data we observed also seem to agree with this theory.

We have found two ways to do the simulation. The first one is the level-set method, which is a general approach to numerical computation for the motion of interfaces. The level-set method uses a continuous scalar field uj for each grain j to track the location and evolution of the interface between different phases or materials [9]. The surface of grain j is where uj=0. uj follows the equation below to evolve over time:

$$\frac{\partial u}{\partial t} + \eta(x,t)\gamma_j(\frac{\nabla u_j}{|\nabla u_j|})|\nabla u_j| = 0 \tag{1}$$

where $\gamma_j(\frac{\nabla u_j}{|\nabla u_j|})$ is the velocity of facet growth, and $\eta(x,t)$ is a function for growth termination where

different grains meet. The level-set method can be implemented by solving the PDE numerically. However, this method has the following drawbacks:

1. Solving the PDE is very demanding on the computing resources.
2. The simulation quality is heavily affected by the spatial and temporal resolution. resolution.
3. The size and shape of each grain needs to be defined at the start of the simulation in order to start growing, which brings limitations to the nucleation assumptions.

The second method is to treat the growing grains as a set of surfaces [10]. This simulation can be implemented by solving a series of linear algebraic equations, which is expected to be more efficient. Moreover, this method can be a continuous states, discrete time (CSDT) or discrete event simulation (DES) system, which does not need discretized space, or time and space, giving more accuracy with less computational resources. On top of solving the equations of grain surfaces, the authors used grain intersection nodes and simulated how they evolve with height to further simplify the computing process. However, this simplification ignores the temporal order of the grains grown. Therefore, we need to adjust this method to simulate the thin film growth at each time step rather than height step. Since we will compare the simulated results to the in-situ XRD data, we need to inspect the volume of each grain throughout the simulation, and calculate the distribution of overall orientation in the polycrystalline film grown.

4 Conceptual model

4.1 Surfaces and intersections

A growing surface $(\mathbf{r}, t)$ can be described as:

$$\mathbf{n} \cdot (\mathbf{r} - \mathbf{r_0}) = v_n(t - t_0), \tag{2}$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the normal of the plane, $\mathbf{r_0}$ is the center (nucleation position) of the grain, t_0 is the nucleation time of the grain, v_n is the growing speed of the plane at the normal direction. Note that the substrate is a special surface whose velocity is 0. The intersection of two surfaces A and B can be described by solving the following equations:

$$\begin{aligned} \mathbf{n_A} \cdot (\mathbf{r} - \mathbf{r_{A0}}) =& v_{An}(t - t_0), \\ \mathbf{n_B} \cdot (\mathbf{r} - \mathbf{r_{B0}}) =& v_{Bn}(t - t_0). \end{aligned}$$

Deriving from the equations above, we have:

$$N_{AB} \cdot \mathbf{r} - b_{AB} = 0, \tag{3}$$

where $N_{AB} = (\mathbf{n_A}, \mathbf{n_B})^T$, and $b_{AB} = (v_{An}(t - t_0) + \mathbf{n_A} \cdot \mathbf{r_{A0}}, v_{Bn}(t - t_0) + \mathbf{n_B} \cdot \mathbf{r_{B0}})^T$. Similarly, for the intersection point of three surfaces can be described as:

$$N_{ABC} \cdot \mathbf{r} - b_{ABC} = 0. \tag{4}$$

Combining $\mathbf{r}$ and t in a 4-dimensional spacetime vector $r^\mu = (x, y, z, t)$, Equation 2 can be rewritten as:

$$v^\mu \cdot (r^\mu - r_0^\mu) = v^\mu \cdot r^\mu - b^\mu = 0, \tag{5}$$

where $v^\mu = (n_x, n_y, n_z, -v_n)$. Intersections can also be written in this form:

$$V_{AB}^\mu \cdot r^\mu - b_{AB}^\mu = 0 \text{ for intersection edges}, \tag{6}$$

$$V_{ABC}^\mu \cdot r^\mu - b_{ABC}^\mu = 0 \text{ for intersection nodes}. \tag{7}$$

This formulation is convenient since we do not need discrete time and calculate intersection for each time step anymore. By solving the 4-vectors, we can directly calculate intersection for any arbitrary time, and thus improving computing efficiency and avoiding loss of accuracy due to the discrete time.

This formulation gives us the power to calculate *turning points* in the crystal growth. When four surfaces coincide at one spacetime coordinate, this coordinate must be a *turning point*. A *turning point* is where and when the growing pattern changes. For example, when a grain is completely "wiped out" from the exposed surfaces, a new intersection point of the three remaining surfaces forms and grows with time. A *turning point* can be determined with the following equation:

$$V^{\mu}_{ABCD} \cdot r^{\mu} - b^{\mu}_{ABCD} = 0 \tag{8}$$

At a *turning point* (x, y, z, t), old intersections may cease to exist, while new intersections may form.

4.2 Evolution

We have three obvious problems to solve to make the simulation working:

1. determine which (living) surfaces are adjacent to each other,
2. determine which (living) surfaces will coincide at a *turning point*.
3. determine the new growth pattern when a *turning point* is found.

For the first problem, an intuitive method to solve this is to use a graph `G` to describe the adjacent surfaces. Initially, each grain is not touching one another, so the adjacent surfaces will be one of the following cases:

1. the two surfaces are adjacent faces in the same grain,
2. one of the surfaces is the substrate.

However, it cannot be said that if two surfaces are one of the cases above, they must be adjacent. The substrate may completely engulf the edge connecting the two surfaces originally adjacent in the grain, making them no longer adjacent.

We decided to use "intersection nodes" as the nodes of the graph, and "intersection edges" as the edges of the graph. The details are discussed in Section 5.4

The second problem is more tricky, as there are multiple possibilities that could cause a *turning point*. For now, we are only focusing on the "terrain" cases where one of the four planes is adjacent to all three other planes. The other types of *turning points* are "aerial" as the grains would "collide" in midair, which are rare, but possible. The "aerial" types can be recovered with "terrain" cases by tracing back in time, which is feasible in our simulation method. At this point, we will only focus on "terrain" cases for now and patch up the "aerial" cases later.

To find the *turning points*, we can investigate each surface, and calculate the potential *turning points* for every combination of three of the adjacent surfaces and the surface itself. Although our system is four-dimensional and time does not seem to be a problem, the events still have their order. Therefore, we need to make a priority queue (earliest time) of the potential *turning points* to find out the next *turning point*. Some *turning points* may affect the other potential *turning point*, while others may not. It might be useful to apply the "light cone" here, to reduce computation complexity: if two *turning points* are out of the "light cone" of each other, then they are not affecting each other. The potential *turning points* outside the "light cone" of the current *turning point* need no updates, while the potential *turning points* inside the "light cone" need to be recalculated using the new graph `G`. "speed of light" can be defined as $\sqrt{\frac{4}{3}v_{100}^2 + v_{002}^2}$.

The third problem first requires identification of the adjacency type of the four planes intersecting at the *turning point*. There are three possibilities: 1. all planes are adjacent to each other; 2. there are two and only two planes that are not adjacent to each other; 3. there is one plane only adjacent to the investigated plane. For any possibility, the graph `G` will be updated accordingly, and the possible *turning points* will be recalculated. We will defer the detailed discussion to Checkpoint 2 or final report due to the space limitation.

4.3 Surface area and volume calculation

To relate the simulation results to the experimental data, we need to calculate the exposed surface area for (100) and (002) surfaces, and the total volume of each grain. The total volume of a grain can be calculated by an integral of the growing surface area over time:

$$V_{\text{grain}} = \sum_{\texttt{all surfaces}} \int_0^t S_n v_n dt \tag{9}$$

If we could find the analytical expression of $S_n(t)$ (which is expected to be a segmented linear or polynomial formula), we will also be able to find an analytical expression of V_{grain} using the integral above. Therefore, the problem reduces to finding the expression of $S_n(t)$ for each surface.

Using the graph described in Section 4.2, we can easily find the adjacent surfaces of the surface under investigation. Using Equation 6 and 7, we can calculate the borders and vertices of the surface. The surface area can then be calculated by dividing them to triangles and calculate the area of each triangle using cross product. By using cross product, not only can we directly calculate the area with 3D coordinates, we also handle non-convex polygons.

In this report, surface area and volume calculation is not implemented due to limited time. The simulation results are visualized by drawing the 3D surfaces using `mpl_toolkits.mplot3d`. Calculating surface area and volume is highly recommended as it relates to material characterization methods, and quantifies the evolutionary selection effect.

4.4 Growth rate scaling

Section 4.1 discussed the case where the surface growth rate v_n are kept constant. However, in an actual electrodeposition process, the surface growth rate is influenced by various parameters, making it much more complicated than the hypothesis presented here.

In galvanostatic electrodeposition, the deposition rate is controlled by the current density. This means that a constant current applied to the system will result in a constant amount of metal ions being reduced and deposited on the substrate surface within a specific unit of time.

Although v_n is varying with the total exposed surface area, we may assume that the ratio between the growth rates in (100) and (002) surfaces are kept the same. We can keep v_n constant throughout the simulation, while logging the history of exposed surface area to restore the growth rates. To restore the growth rates are of the same effect with scaling time. Therefore, we can restore the real time interval dt' by scaling the simulation time interval dt:

$$dt' = \frac{\mathbf{S_n}(t) \cdot \mathbf{v_n}}{D_t V} dt, \tag{10}$$

where $D_t V$ is the total volume change rate, which is kept constant in electrodeposition, $\mathbf{S_n}$ is the vector consisting total exposed area of each type of surfaces, and $\mathbf{v_n}$ is the vector consisting corresponding simulation growth rates.

In this report, growth rate scaling is not implemented due to limited time.

5 Simulation

5.1 Platform

The simulation will be developed using `python`. Packages include: `Numpy`, `Scipy`, `NetworkX` `Matplotlib`, `mpl_toolkits.mplot3d`, `itertools`, and `queue`.

5.2 Simulator

The simulator is designed in a function oriented fashion. The simulation process is done in three steps: 1. nuleation simulation; 2. initial collision calculation; 3. crystal growth simulation. Nuclei positions are

generated by a random variation from an m-column, n-row lattice. The probability of seeing a nuclei is set to be uniform over the defined substrate area. Nuclei orientations are uniformly random. Initial collision events are calculated with linear programming, and are inserted into the event handler.

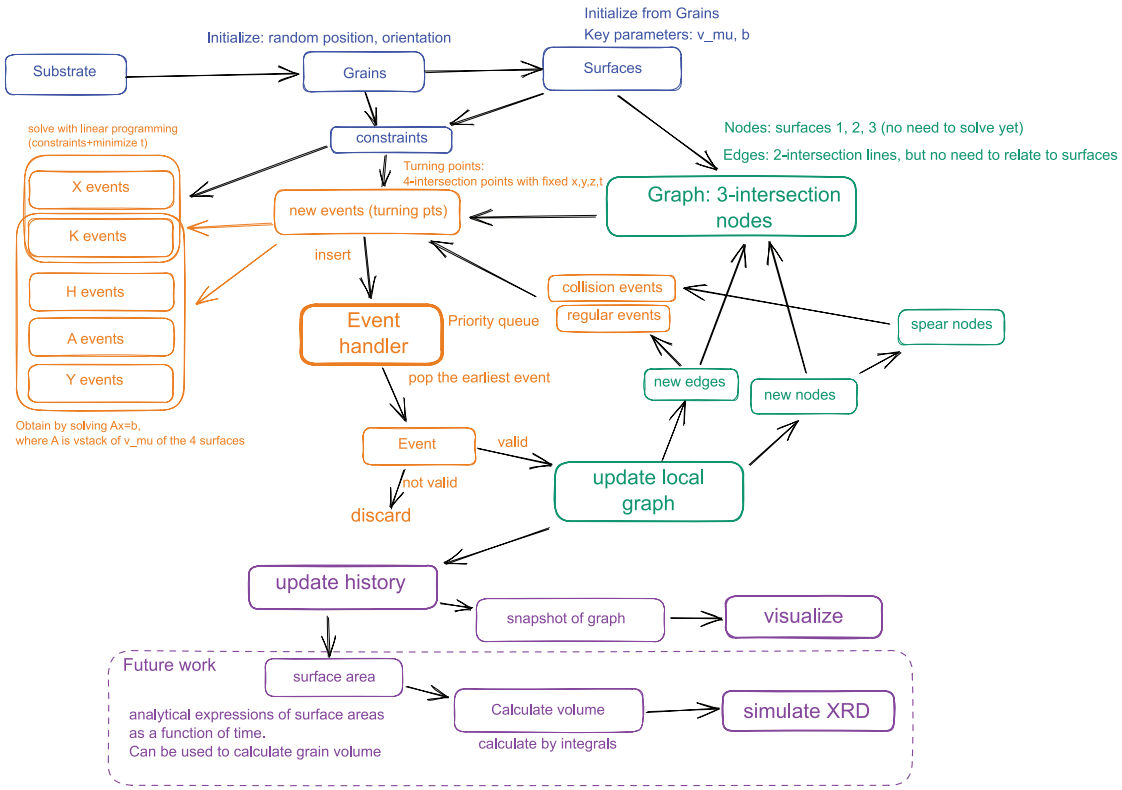


Figure 2: Simulation code design schematic.

After initializing the event handler with collision events, the simulator calls "handle_next" function repeatedly to simulate crystal growth step by step. The algorithm goes through a cycle as demonstrated in Figure 6.

5.3 Surface and grain

The class "Surface" is responsible for handling each surface in the simulation, including the exposed surfaces and the substrate. "Substrate" and "Surface_Hexagonal" inherits "Surface" for construction of different types of surfaces. A Surface object contains $v^\mu = (n_x, n_y, n_z, -v_n)$ and $b = \mathbf{r_0} \cdot \mathbf{n}$ that describes an infinite plane that moves with time. v^μ and b does not change over the entire simulation process. An id attribute defines what type of surface it is (e.g., id=-1 for the substrate). An intersection function calculates the intersection of four surfaces, where a turning point (event) should happen. A function to calculate the position of intersection nodes at a certain time is also implemented.

The class "Grain" is responsible for handling the initialization of the surfaces, and resolving the initial intersection of the grains with the substrate. As stated in Section 4.2, the handling of initial adjacency between surfaces is non-trivial. To avoid connecting surfaces that are not adjacent, the position of each intersection node at time=1 is calculated, and determine if it is outside any of the grain surfaces. If so, the intersection node is excluded from the graph.

5.4 Intersection graph

The class "IntersectionGraph" is responsible for constructing the initial graph and continuously updating the graph as simulated events take place. During initialization, nodes from the nucleated grain surfaces are

added to the graph, as well as the edges connecting them once detected. Subsequently, the decision to the add and remove nodes and edges are executed in "update_graph" based on the type of event processed in the event handler. The process is implemented using NetworkX graph methods.
Based on the input containing pre-event nodes and post-event nodes, "IntersectionGraph" will find the location of this event and all associated graph elements. There are certain events that the outcome is geometry-dependant and not deterministic based on graph relations along such as when one node collides with a edge (K event) or two edges contact each other (X event). In such instances, the code will try to solve for the location of the relevant nodes explicitly and determine their connection configuration.

5.5 Event handler

The class "Event" is responsible for constructing events for the event handler. Each event is a turning point where four surfaces intersect at a unique coordinate in the 4-dimensional space-time.

The class "Event_handler" is responsible for managing the event queue. This class inherits "Priority Queue" from the package "queue". The event handler always handle the event with the earliest time. First, it will check if the event is valid (e.g., if the nodes required are still in the graph). Then it calls functions in the "Intersectiongraph" to update the graph according to the characteristics of the event, construct new events from the new edges, and put the new events in the event queue.

5.6 Visualization

The simulation results for Zn crystals with different orientations were plotted in a 3D space to see how random-oriented Zn nuclei form/evolve different crystal orientations as a function of time. Each surface was created by connecting its vertices in order to represent the crystal structure. To differentiate between crystal planes, (002) and (100) planes were represented with red and green color, respectively.

5.7 Verification

To verify that the implementation follows our conceptual model, test cases were studied using the 'test.py' script with few number of grains and steps. By doing so, we can observe whether the graph elements (grains, surfaces, nodes and edges) of the simulation evolve as expected. Most importantly, we paid close attention to the occurrence of event at each step to examine whether they are captured and categorized by the event handler and processed properly in the intersection graph. If an event is found to unfold in disagreement with our understanding, the implementation will be either revised to correct the error or patched as a new type of event to address the special cases. For example, the situation can be complicated when a node coincides with an edge. The original edge must be removed and be replaced with two new edges connecting the correct nodes. We have solved the "incident edge" issue for most cases, but there are still rare events where our patches fail due to a obscure issue associated with surface identification.In such incidences, another edge on the same surface will also be treated as the incident edge, resulting in a partially cropped surface.

6 Experimental results and validation

6.1 Nucleation simulation

During the nucleation phase, the crystals are expected to initiate their growth on the substrate in a macroscropically uniform distribution with random orientations. As shown in Figure 3, the simulation generates grains of random orientations on top of the stationary substrate as expected. To introduced an appropriate amount of randomness to he distribution, the substrate are divided by cells of the unit size where only one grain could nucleate but may be centered at any location within its boundary, leading to a locally random nucleation map. Because Zinc crystal is the model material for this study, HCP was choosen as the geometry for the grains. Overall, we believe the nucleation simulation is a accurate representation of our conceptual understanding of the system and a possible predecessor to the electrodeposited polycrystalline

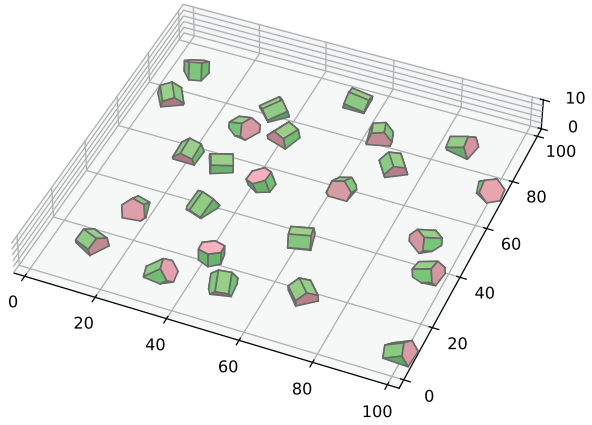


Figure 3: Nucleation simulation with a 5 by 5 grid.

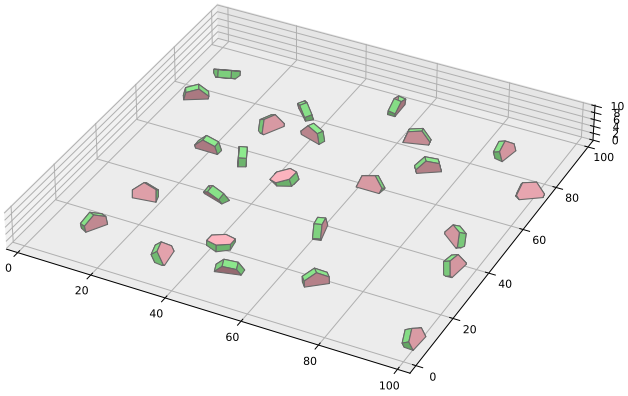
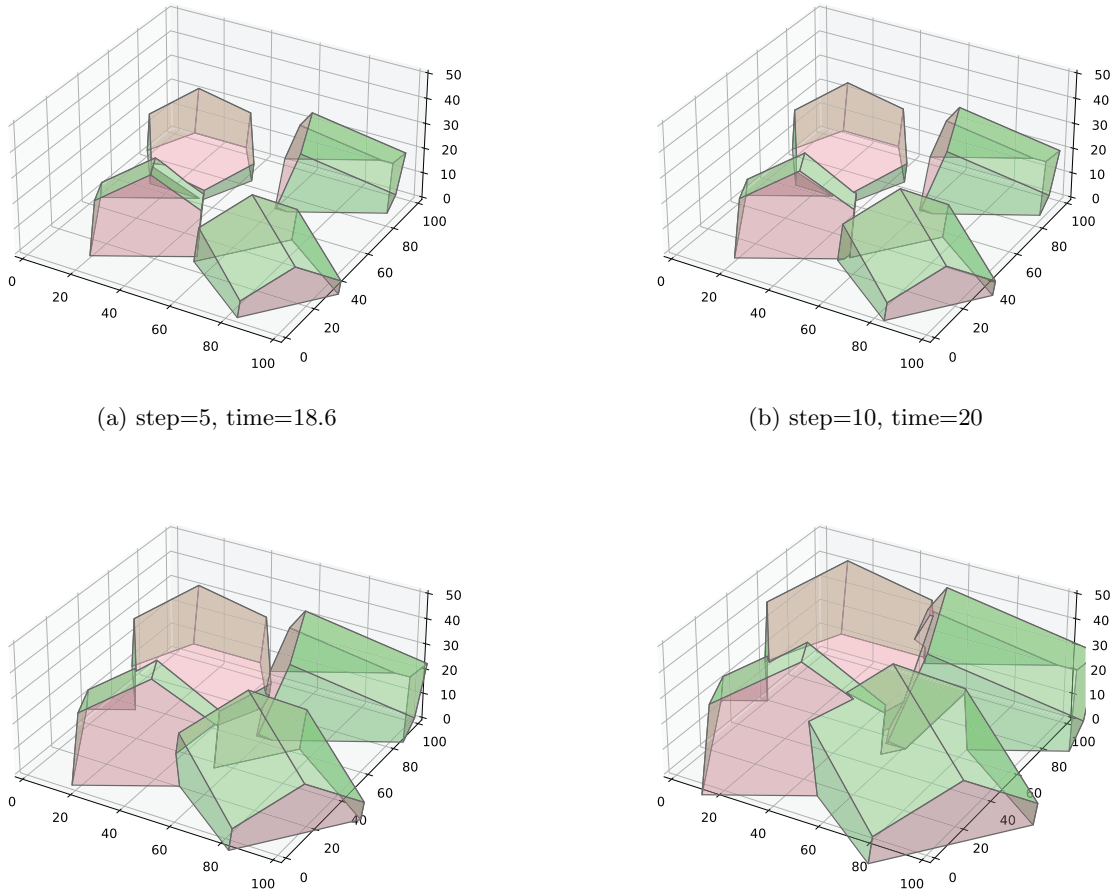


Figure 4: Nucleation simulation with 3x (100) phase growth speed.

Zn shown in Figure 7.
In addition, the physical parameters of the grain including cyrstal structure and growth speed can be controlled in the 'simulation_parameters' file. In reality, different phases of the crystal often possess unique physcial properties and exhibit different behavior during its growth. Figure 4 presents the same nucleation process with the grains growing three times as fast in the [100] direction. The change of growth speed leads to significantly reduced grain height and enlarged [002] surfaces shown in red. This initial state more representative for some grains with high aspect ratio.

6.2 Crystal growth simulation

After the nuclei stage, the formed nuclei begin to grow in the following deposition process. Initially, there is still large space between these nuclei and the formed nuclei continue to grow while maintaining their geometry. After growing for some time, these nuclei grow into small Zn grains and begin to make contact with each other. The outcome of their interactions are determined by the "turning point" as discussed in our conceptual model.



(a) step=5, time=18.6

(b) step=10, time=20

(c) step=15, time=24

(d) step=22, time=30

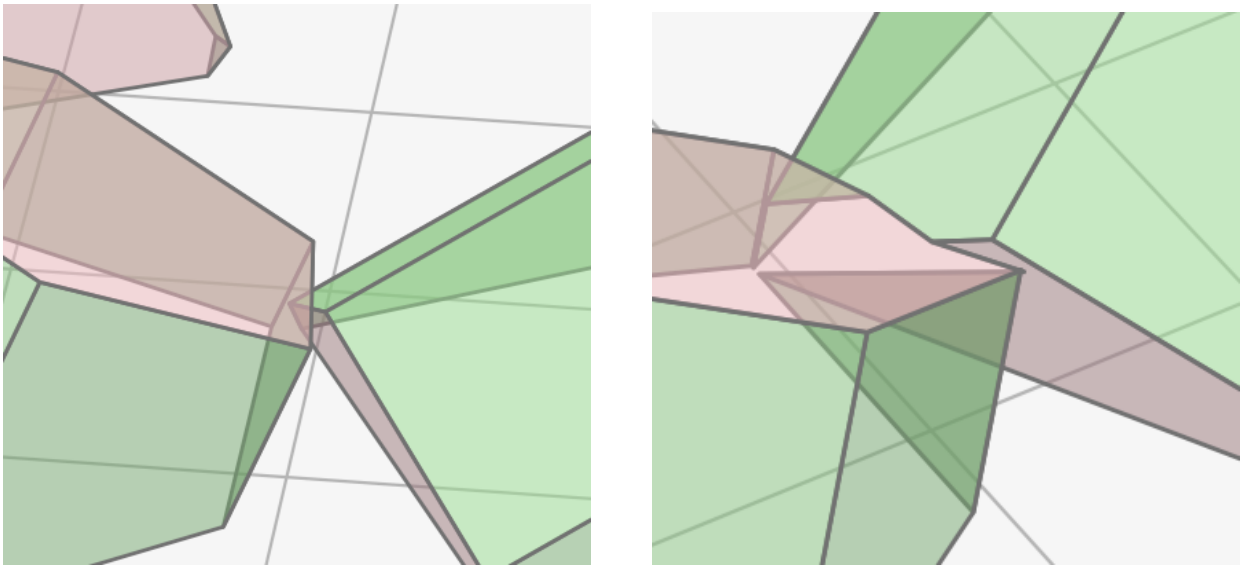
Figure 5: Crystal growth simulation with 4 grains

In the simulation process, the intersection nodes and edges will experience different types of events where their linear behavior has a sudden change. The types of events can be categorized as below:

- X event: two independent edges collide in midair, producing 4 new nodes.
- K event: an intersection node runs into an incident edge or incident surface, producing 3 new nodes.
- H event: two intersection nodes with a common edge merge, producing two new nodes with a common edge.
- A event: three intersection nodes with common edges between each pair merge, producing 1 new node, and 1 surface stops growing.

- Y event: four intersection nodes with common edges between each pair, and all nodes, edges, and surfaces stop growing.

Since the surfaces are moving at a constant speed, the behavior of the intersection nodes and intersection edges is predictable. Here, we take X event as an example. As shown in Figure 6, two independent edges coincide at the beginning of the X event. In an X event, initially there are no nodes involved in the event. After the event happens, four new intersection nodes will appear, because there are $C_4^3 = 4$ ways to select three surfaces out of 4 that are involved in the event. Two nodes will be on each initial edge, and there will be an edge between the new nodes if they are on different edges. Each initial edge will be replaced by two edges defined by the original nodes of the edge and the two new nodes on the edge. Figure 6 shows the situation after the X event (Note that one of the new nodes moves very quickly due to the small angle between the two grains, and has gone out of the edge, forming another event).



(a) Start of the "X event"

(b) Intersection after the "X event"

Figure 6: Close-up at an "X event".

6.3 Comparison between simulation results and experimental data

From simulation results, we see how the Zn nuclei first grow into small crystals and then interact with each other. Figure 7 shows the morphology of the electrodeposited Zn nuclei and the morphology after their growth interactions. From these images, we see the Zn nuclei have regular hexagonal shapes and are fairly uniform, which is in agreement with the simulated nucleation results. After these Zn nuclei make contact with each other, their growth mode is impacted by the surrounding, resulting in bulky agglomerate of irregular shapes. This can be partially observed underneath the small grains on the top surface in the SEM image and can be attributed to the grain interaction during crystalline growth as shown in Figure 5. Overall, the simulation results and experimental data are consistent with each other.

7 Conclusion

In this project, we have developed a multicrystalline growth simulation method using discrete event simulation (DES). We designed a novel method to describe the crystal growth system with 4-dimensional space-time vectors. The simulation is built around *turning point* events, where 4 surfaces coincide at 1 unique point in the 4-dimensional space-time. The status of the system between events is described by an "intersection graph", where the nodes are intersection nodes defined by 3 surfaces, and the edges are

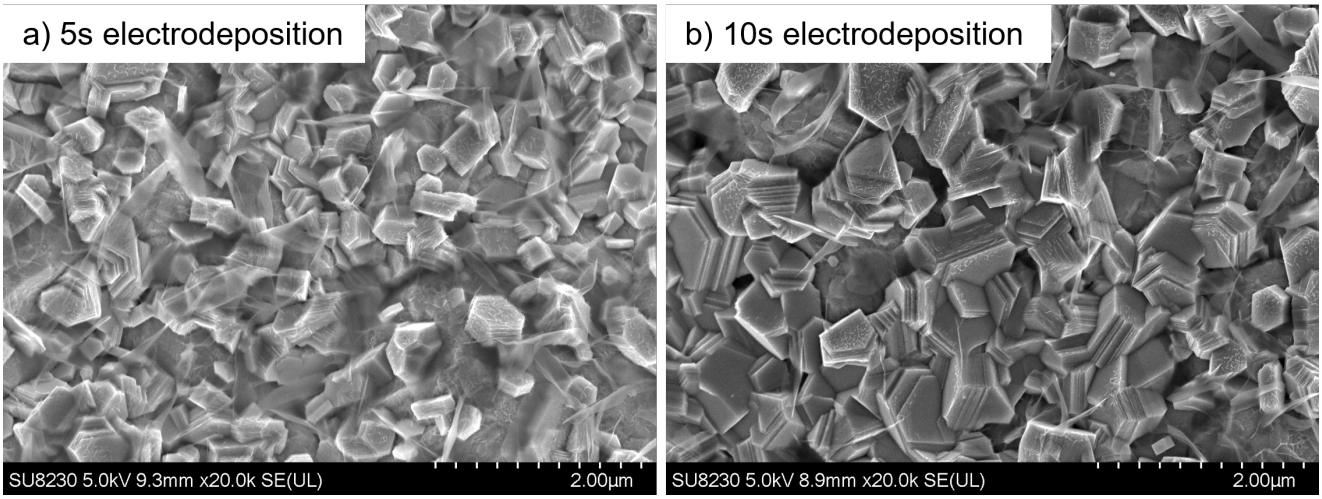


Figure 7: Morphologies of electrodeposited Zinc after 5 seconds a) and 10 seconds b).

intersection edges between 2 surfaces. The simulation is conducted in a time-forward fashion, while we also explored the possibility to go back in time. The simulation results have shown that our novel method is valid for different types of *turning point* events, and predicted the evolution of the intersection edges and intersection nodes.

Due to limited time and resources, some aspects of this study were not fully developed and could benefit from further investigation. The main limitation of the simulation at the current stage is handling all types of events as the geometry of the grains gets more complicated over time. Even though most events are correctly handled, there are always occasional cases where the nodes and edges are not updated correctly, and these error propagates as the simulation goes on.

Regarding to the analysis of this simulation, computing the surface area and volume for the surface and grain system would be valuable because they will enable us to measure some important material properties of the system such as the X-ray diffraction (XRD) pattern and surface roughness that can further validate the simulation. Also, this simulation considers the simple case where nucleation only takes place at the beginning on the substrate and each phase of the grain facets grows at constant rate, while in reality new nuclei are formed continuously at the solid-liquid interface, and the individual growth rate is variable and scales depending on surface exposure. Implementing a time/current-dependant nucleation model and exposure-based growth function can significantly improve the capability of the simulation to model real electrodeposition systems.

References

[1] Chan Pei Yi and Siti Rohana Majid. The electrochemical performance of deposited manganese oxide-based film as electrode material for electrochemical capacitor application. *Semiconductors-Growth and Characterization*, page 1193, 2018.

[2] Wentao Yuan, Xueyu Nie, Guoqiang Ma, Mengyu Liu, Yuanyuan Wang, Shigang Shen, and Ning Zhang. Realizing textured zinc metal anodes through regulating electrodeposition current for aqueous zinc batteries. *Angewandte Chemie*, 2023.

[3] Jingmin Zhang, Weiwei Huang, Longwei Li, Caiyun Chang, Kai Yang, Lei Gao, and Xiong Pu. Nonepitaxial electrodeposition of (002)-textured zn anode on textureless substrates for dendrite-free and hydrogen evolution-suppressed zn batteries. *Advanced Materials*, page 2300073, 2023.

[4] Mie Andersen, Chiara Panosetti, and Karsten Reuter. A practical guide to surface kinetic monte carlo simulations. *Frontiers in chemistry*, 7:202, 2019.

[5] Michael W Anderson, James T Gebbie-Rayet, Adam R Hill, Nani Farida, Martin P Attfield, Pablo Cubillas, Vladislav A Blatov, Davide M Proserpio, Duncan Akporiaye, Bjørnar Arstad, et al. Predicting crystal growth via a unified kinetic three-dimensional partition model. *Nature*, 544(7651):456–459, 2017.

[6] Adam R Hill, Pablo Cubillas, James T Gebbie-Rayet, Mollie Trueman, Nathan de Bruyn, Zulaikha Al Harthi, Rachel JS Pooley, Martin P Attfield, Vladislav A Blatov, Davide M Proserpio, et al. Crystalgrower: a generic computer program for monte carlo modelling of crystal growth. *Chemical science*, 12(3):1126–1146, 2021.

[7] Kenneth G Libbrecht. Physical dynamics of ice crystal growth. *Annual Review of Materials Research*, 47:271–295, 2017.

[8] A Van der Drift. Evolutionary selection, a principle governing growth orientation in vapour-deposited layers. *Philips Res. Rep*, 22(3):267, 1967.

[9] Peter Smereka, Xingquan Li, Giovanni Russo, and David J Srolovitz. Simulation of faceted film growth in three dimensions: microstructure, morphology and texture. *Acta Materialia*, 53(4):1191–1204, 2005.

[10] X Jiang, G Yu, and ST Lee. Simulation of polycrystalline 3d film growth: An investigation of the evolution of grain size and texture in diamond films. *Applied Physics A*, 74:217–224, 2002.

A Division of labor

1. Model building - Zhaonan Liu
2. Simulation coding - Zhaonan Liu, Shikai Jin
3. Simulation execution - Shikai Jin, Zhaonan Liu
4. Data visualization - Yifan Ma
5. Experiments verification - Yifan Ma
6. Report drafting - Shikai Jin, Yifan Ma, Zhaonan Liu