Directed Self-Assembly:

A Study of the Effect of Electric Fields on Silica Monolayers

Matthew O. Withers

A thesis presented in partial fulfillment of
the requirements for the degree of
Bachelor of Science

Department of Physics and Engineering
Washington and Lee University
Lexington, VA, USA
May 19, 2020

Abstract

In this thesis, we describe, analyze, and extend ionic self-assembly of monolayers (ISAM), a bottom-up nanostructure production technique designed to coat surfaces in uniform layers of charged nanoparticles. Using mean field theory, we develop cooperative sequential adsorption with evaporation (CSAE) models of the assembly process, designed to predict the particle coverage density of ISAM samples. We simulate the particle assembly process via the Monte Carlo technique, and we evaluate our CSAE models primarily by comparing them to these simulated results. Finally, aided by scanning electron microscopy, we analyze experimental ISAM samples. This experimental approach provides us with information about the time scale of assembly, as well as the relationship between our CSAE models and particle suspension concentration. Our approach considers ISAM under no external influence, as well as ISAM conducted under constant and oscillating electric fields. Assembly under electric fields represents a type of directed self-assembly of monolayers (DSAM), an emerging technique designed to control particle coverage density using an external influence.

Acknowledgments

I would first and foremost like to thank my thesis adviser, Dr. Dan Mazilu, for all of his assistance, not only in terms of the writing of this thesis, but also in terms of the academic, professional, and personal guidance he has offered throughout my undergraduate career. His advice in the laboratory and on trips to conferences both at home and abroad have had a significant impact on my development as a scientist and communicator.

I would also like to thank my academic adviser, Dr. Irina Mazilu, for all the assistance she has offered in terms of planning my academic trajectory. I would also like to thank her for taking the time to mentor me in teaching and scientific communication. These skills will be invaluable throughout the rest of my career. Furthermore, this thesis would not have been possible without the countless hours of theoretical discussions we shared.

In the laboratory, I would like to thank Ms. Emily Falls for her assistance with scanning electron microscope training and certification, as well as Mr. David Pfaff for answering any question I had concerning the equipment in the IQ Center. I would also like to thank Mr. Chris Compton for all of his laboratory expertise, especially with respect to the acquisition of new, quality laboratory equipment.

Finally, I would like to thank all of my student partners in the nanoscience laboratory for providing a fun and collegial work environment. Partners over the years include Elise Baker, Gillen Beck, Laura Bruce, Emily Hassid, Cory Morris, Mitchell Roberts, Ben Zeman, and Nolan Zunk.

Some portions of the research contained in this thesis was completed as a part of Washington and Lee University's Summer Research Scholars Program. My stipend for the program was provided by the E. A. Morris Charitable Foundation. I presented much of this research at conferences, including the APS March Meeting (2019) in Boston, MA, and the 8th International Conference on Mathematical Modeling in Physical Sciences in Bratislava, Slovakia. Travel to these conferences was funded, in part, by the Johnson Scholarship Program.

Contents

Chapter 1

Introduction and Background

1.1 Introduction to Nanoscience

On December 29, 1959, renowned physicist Richard Feynman addressed members of the American Physical Society at their annual gathering. His talk, entitled "There's Plenty of Room at the Bottom" [1], considered a world in which scientists could carefully manipulate individual molecules and atoms to store information on the smallest scales. For nearly two decades, his thoughts went mostly unnoticed. However, by the 1980s, they had become an important tool for members of the newly emerging nanoscience community, which used Feynman's ideas to both inspire new areas of research and justify their belief that meaningful physics could be reasonably conducted at the nanoscale.

Feynman's exploration of what would become nanoscience begins with an unusual question: is there enough room on the head of a pin to print the entirety of the *Encyclopaedia Britannica*? He answers this question by stating that the head of a pin would need to be magnified roughly 25,000 times before its area would equal the area of all the pages in the encyclopedia. Naturally, therefore, one could print the entire encyclopedia on a pin head if he or she could simply reduce the text of the encyclopedia by 25,000 times. Feynman argues that, even at such a small scale, the smallest discernible dots contained in a standard printing of the encyclopedia would contain around 1000 atoms—more than enough to ensure that the information could be preserved for a long period of time without corruption. Furthermore, Feynman notes that the electron microscopes available in 1959 could certainly read text of that size, and an inverted electron microscope lens could be used to focus ions onto the pin's surface to engrave the text. Once one copy of the text had been created, further copies could easily be produced via a plastic mold of the original.

Feynman then considers the physical storage of information via a binary code. Reasoning that each letter within the Latin alphabet would require between six and seven bits of data for unique storage, he suggests using cubes of $5 \times 5 \times 5 = 125$ atoms to encode the information within (not just on the surface of) a pin head. For one binary state, called a dot, one type of atom would be used, while for the other state, called a dash, another type of atom would suffice. Given these parameters, Feynman calculates that "all the books in the world can be written in this form in a cube of material one two-hundredth of an inch wide—which is the barest piece of dust that can be made out by the human eye." Herein lies Feynman's central argument: not only does the nanoscale provide *sufficient* room to store vast amounts of information, but also it appears to provide *plenty* of room.

After demonstrating the vast storage capabilities of the nanoscale, Feynman continues his argument by taking up a discussion of the electron microscope. While avoiding a discussion of any practical engineering details, Feynman suggests that the electron microscope of his day could be improved to the point of being able to discern individual atoms. His reasoning, he states, relies upon the fact that the wavelength of an electron is only around 1/20 of an angstrom, much smaller than the width of an atom. The results of improving the resolution of the electron microscope to even 100 times its 1959 resolution would have numerous consequences for the scientific community. Biologists would be able to probe the structures of DNA, RNA, amino acids, proteins, and microsomes directly. Chemists, rather than using an extensive set of reactions to determine the atomic makeup of a complicated molecule, could simply look at the molecule under the electron microscope to determine its structure and constituent parts.

Inspired by the ability of biological systems to complete complex tasks at the molecular level, Feynman continues by considering a variety of miniaturized machines and devices. He first suggests a miniature computer, perhaps one with wires 10 to 100 atoms in diameter, which would allow for circuits thousands of angstroms in width. From Feynman's point of view, developing miniaturized computer components is an important step in creating computers powerful enough to complete human-like tasks, such as (what we now call) machine learning and image recognition. In 1959, standard computers took up entire rooms. Building a computer with enough processing power to learn from past experience and recognize images would take up a space roughly the size of the Pentagon. A computer of this size would encounter a vast array of problems besides simply being unwieldy. First, it would require too much material—not just in terms of cost but, more importantly, in terms of availability on Earth. Second, it would produce too much heat. Third, it would consume as much power on a yearly basis as is produced annually by the Tennessee Valley Authority. Finally, because information would need to be passed between components separated by vast distances, such a computer would be prohibitively slow.

Feynman considers the production of other miniaturized machines via a variety of methods. First, he suggests evaporating alternating layers of conductors and insulators onto a surface, thus converting the surface into a viable circuit containing components as complicated as coils, capacitors, and transistors. He also suggests that the mechanical devices of our world (e.g., cars and engines) could be scaled down directly to nanosize. Doing so would require considerations of force scaling and material strength. Furthermore, any electrical systems in the scaled-down device would need to be redesigned since magnetic properties change at such a size. However, scaled-down devices would have a variety of benefits, the chief of which would be the elimination of a need for lubricating agents and heat sinks (devices on the order of 10^{-9} m would dissipate heat much too quickly to require standard cooling techniques). To produce such scaled down devices, Feynman imagines a chain of tiny hands, each a quarter of the size of the one before it, which eventually produce the "tools" necessary to manufacture on the nanoscale. The engineering feasibility of this idea is less important than its spirit: Feynman imagines a world in which we produce versions of macroscopic tools which function on the nanoscale, a concept which would allow for advancements in fields as far reaching as medicine and electrical engineering.

Feynman ends his discussion of the nanorealm by considering the potential to produce synthetic molecules directly. If an electron microscope could (at least theoretically) view individual atoms, why could some other tool not be used to move them around and arrange them in a desired structure? Such a tool would have a profound impact on the field of chemistry. No longer would chemists have to rely upon long reaction chains just to produce a sample of a molecule that inevitably contains impurities. Instead, if a chemist desired a particular molecule, he or she could simply put it together piece by piece.

The impact of Feynman's speech on the field of nanoscience is perhaps best measured by the number of real-world advancements it has inspired. Electron beam lithography [2, 3, 4, 5] is an excellent example. In electron beam lithography, a beam of electrons is focused onto the surface of an electron-sensitive film known as a resist. When the beam makes contact with the resist, it changes its properties ("cures" it). Later, a solvent can be applied to the surface of the resist, causing either the cured or uncured portion to fall away. In this way, intricate patterns and structures can be drawn on the resist's surface. The resist is then easily used as a template for etching onto a more solid material (substrate) such as a silicon wafer. Electron beam lithography is very similar to Feynman's suggestion that text from the *Encyclopaedia Britannica* could be written onto the head of a pin by reversing the optics of an electron microscope. Perhaps even more directly inspired is nanostructure etching via focused ion beams [2, 6, 7]. A focused ion beam is simply a steady stream of ions focused via an electromagnetic optical system until powerful enough to cut away a surface. Rather than relying upon resists and solvents, focused ion beam etching cuts nanostructures into a substrate directly, allowing for rapid prototyping of complex nanostructures.

Feynman's discussion also predicted the development of nanostructure stamping technology [2, 8, 9, 10]. During his discussion of the encyclopedia, Feynman suggests creating plastic molds of its miniaturized version so that copies could be easily produced for students and researchers all over the world. George Whitesides' nanostamping technology [8], in which silicone rubber is allowed to cure on top of an already-produced nanostructure, is almost identical to Feynman's vision.

While three-dimensional, high-density storage of the type Feynman envisioned when discussing his attempt to compress every known book into a single speck of dust has not been realized, nanoscientists have managed to utilize three dimensions in the construction of nanostructures. Perhaps the best known example of a three-dimensional nanostructure is the integrated circuit [11, 12]. Incorporating potentially thousands of transistors, resistors, capacitors, coils, and other electrical components, integrated circuits require more space than the simple two-dimensional plane to be realized.

Feynman's suggestions about imaging in biology and chemistry have ultimately been realized via the atomic force microscope [13], which routinely produces images of individual atoms [14]. The atomic force microscope can even be used to detect the atomic structure of molecules [15] and move atoms around [16], allowing for the production of fully-synthetic molecules.[1] Furthermore, his ideas concerning the production of nanoscale computers and machines, particularly his comments on nanostructure production via evaporation, have found life via the study of biological motors [17] and the development of molecular beam epitaxy [18].

The techniques for producing nanostructures in today's research environment are generally divided into one of two categories. *Top-down* approaches, as the name suggests, involve the selective removal of material from a surface until all that remains is the desired nanostructure [2]. On the other hand, *bottom-up* approaches involve the use of chemical processes to deposit materials sequentially on a surface, eventually building up a nanostructure layer by layer [2]. It is often easy to think of top-down approaches like the work of a sculptor. The sculptor begins with a large block of material and removes portions little by little until a masterpiece emerges. In contrast, bottom-up approaches are similar to the work of a painter, who layers paint on a canvas until a clear, structured image appears.

[1] This method of molecule production is still nowhere near as efficient as typical chemical synthesis methods.

While the methods generally classified as top-down approaches are especially diverse, most laboratories use procedures that include similar steps. Almost every procedure includes an *oxidation* step, in which a protective layer of silicon dioxide is placed on the surface of a silicon substrate. During *masking*, photolithographic methods are used to etch features into the freshly deposited silicon layer. With the desired features etched, thus exposing portions of the silicon substrate, the implantation phase commences. In *implantation*, charged ions are delivered to the exposed portions of the substrate. This leads to doping, which changes the exposed areas' electrical properties relative to the rest of the substrate. After implantation, etching occurs. During *etching*, the manufacturer removes the protective silicon dioxide layer and any undesired potion of the silicon substrate. *Metallization* involves the use of evaporative or electromechanical methods to deposit metals on the surface of the nanostructure, allowing for electrical contacts to form between critical components. Finally, manufacturers use *lift-off* to deposit a photoresist on the surface of the nanostructure, which, when selectively cured and placed in a dissolving bath, allows for control over the selective removal of material [2].

Well-established nanostructure production processes that make use of top-down approaches include photolithography [19], electron beam lithography [2, 3, 4, 5], reactive ion etching [2, 6, 7], molecular beam epitaxy [18], self-assembled masks [20], focused ion beam milling [21], and stamp technologies [8]. Utilizing different combinations of the methods explained above, these manufacturing processes offer nanoscientists a high level of control over the nanostructure's construction.

While each step of nanostructure production via top-down approaches is carefully controlled by the experimenter, bottom-up approaches use a much more hands-off approach, in which the random collisions of suspended particles facilitate the assembly of a structure. Using random collisions as a means of assembly implies that bottom-up approaches rely heavily on the enthalpy and entropy of the system. If the enthalpy is too low, the nanoparticles will not have enough energy to (weakly) bond and form a structure. If the entropy is too low, incorrectly formed nanostructures (often dubbed "erroneous" nanostructures in the literature) will be unable to disassemble, thus ruining the sample. Conversely, if the entropy is too high, any correctly formed nanostructures will be unstable. This instability will result in their quick dissolution [2].

Ideas from traditional chemical synthesis and techniques from contemporary experimental chemistry are also important in bottom-up nanostructure production. For example, the theory of chemical synthesis helps to predict the amount of reactants needed to drive an irreversible reaction. Similar analysis helps nanoscientists determine the amount of nanostructure components to add to a solvent to facilitate the self-assembly of a nanostructure. Concepts from surface chemistry are also especially important in nanostructure production, as nanostructures commonly have high surface to volume ratios, and their surface atoms are often unable to bond traditionally with atoms of the same type [2].

Contemporary examples of bottom-up nanostructure production include organic synthesis [22], vesicle [23] and micelle [24] production, self-assembled molecular monolayers [25, 26, 27], kinetic control of growth [28, 29] (nanowire [30] and quantum dot [31, 32, 33] production), and DNA nanotechnology [34, 35]. In each case, the nanoscientist sacrifices direct control over the construction of the assembly in order to take advantage of self-assembling characteristics.

1.2 Self-Assembly of Molecular Monolayers

One of the most versatile nanostructure production methods currently in use is the self-assembly of molecular monolayers. This method, in which charged particles arrange themselves into particle sheets of uniform thickness, is used to coat objects in nanoparticles, thus changing the objects' surface properties. For example, self-assembly of molecular monolayers has been used to apply thin films to sheets of glass, thus producing the thin-film interference necessary for an anti-reflective coating [36, 37, 38]. For the remainder of this thesis, we discuss self-assembly of molecular monolayers in detail, in particular the experimental techniques necessary to produce high quality monolayers. We also develop and evaluate modeling and simulation techniques designed to predict monolayer properties before the assembly process begins.

A functional method of achieving self-assembly of molecular monolayers was first proposed by Iler [25] in 1966. Building upon the work of Langmuir, whose 1941 patent [26] states that adsorbing ions will form a single layer until every available occupation site is filled, Iler demonstrated how particles of colloidal size can adsorb onto a surface of the opposite charge. Iler's advancement was particular important to nanoscience because most of the particles that scientists wish to employ in the self-assembly process (e.g., SiO_2, TiO_2, etc.) are of colloidal size.

Iler's method began with the selection of an assembly surface. While he stated that any anionic surface is sufficient, his experiments made use of a sheet of clean, hydrophilic glass.[2] The siliceous nature of the glass ensured that it would remain anionic throughout the experiment. With the assembly surface selected, Iler then coated it with a cationic layer. In his case, the cationic layer was a 0.25% aquasol of colloidal boehmite alumina containing 5-6μm (diameter) fibrils of AlOOH. He followed the cationic layer with a 2% aqueous sol of colloidal silica. Each silica particle was essentially spherical with a diameter of approximately 100 nm. Attracted by the cationic aluminum layer, the silica adsorbed onto the surface of the glass, forming a uniform layer, or thin film. Iler found that, after rinsing the assembled layer of silica, he could repeat this process to form additional layers. This permitted to creation of silica films of varying thicknesses.

Iler made two important observations while developing this assembly method. First, he noted that the alternation of cationic and anionic layers was essential to the development of multiple layers of silica. Silica attempting adsorption after a uniform layer had formed would be repelled by the charge of already deposited silica. By placing an aluminum layer on top of the adsorbed silica, the suspended silica would be attracted to the resulting cationic charge, thus permitting the formation of an additional layer. Second, Iler observed that films with thicknesses over 50 nm would become visible in reflected light. This visibility is due to the thickness of the film approaching the wavelength of visible light.

Iler also noted a number of factors affecting the formation of layers. First, he found that rinsing the sample with distilled water between the deposition of cationic and anionic layers was essential to ensuring that the next layer would be uniform. Second, he observed that the concentration of the cationic and anionic colloidal solutions used could also be important. As a general rule, he suggested using concentrations less than 0.5% for small particles (defined as particles with specific surface areas on the order of several hundred square meters per gram) and 3-5% for larger particles (e.g., the 100 nm silica used in his experiment). Third, the pH of the solutions was critical. In particular, Iler found that the adsorption of this silica particles took place most efficiently (i.e., rapidly and completely) in the low pH range (pH 2-4). Finally, suspensions that contained

[2] Iler selected black-tinted glass to aid in the analysis of the optical properties of the assembled layers.

supercolloidal aggregates or a gel phase would produce layers with significant irregularities. These irregularities would often prevent the formation of a layer capable of producing interference, thus rendering the layer invisible when observed under visible light.

Iler's methods were built upon by Lvov, Ariga, Onda, Ichinose, and Kunitake [27] in 1997. Using a quartz crystal microbalance (QCM), scanning electron micoscopy (SEM), and atomic force micoscopy (AFM), their team completed more thorough analysis of monolayers produced using Iler's method.

In their experiments, Lvov et al. used poly(diallyldimethylammonium chloride) (PDDA), sodium poly(styrenesulfonate) (PSS), and poly(ethyleneimine) (PEI) as their bonding agents. They tested a variety of anionic nanoparticles, including silicon dioxide (SiO_2), titanium dioxide (TiO_2), and cesium dioxide (CeO_2). Their primary investigation involved the use of the quartz crystal microbalance technique to detect changes in the mass of the assembly. This mass value could then be converted into a measurement of the thickness (d) of the assembled layer. Their team was interested in determining whether the thickness of the film increased linearly as additional particle layers were deposited. For the particular QCM their team used, the change in quartz frequency depended directly upon the mass (M) and inversely upon the surface area (A) of the sample:

$$\Delta F = (-1.83 \times 10^8)\frac{M}{A}. \tag{1.1}$$

Furthermore, the sample thickness (d) was related to this frequency shift via

$$d \text{ (nm)} = 0.022(-\Delta F \text{ (Hz)}). \tag{1.2}$$

Lvov et al. assembled a thin film layer by layer, measuring the change in frequency it produced within the QCM after each layer (either cationic or anionic) had fully formed.[3] Their results demonstrated several interesting properties. First, the deposition of the bonding agent (PDDA, PSS, PEI, etc.) contributed minimally to the thickness of the film. This behavior is shown by the nearly constant ΔF at odd adsorption steps (see Figure 1.1). Meanwhile, the adsorption of nanoparticles produced a consistent frequency shift, which corresponds to a consistent increase in film thickness, no matter the layer being produced (see even adsorption steps in Figure 1.1). Furthermore, the magnitude of the frequency shift (i.e., the magnitude of the increase in film thickness) depended upon the the concentration of nanoparticles but not on the size of the particle used (see Figure 1.2). Specifically, an increase in concentration corresponded to an increase in the size of the growth step. These results were particularly significant because they indicated that the experimenter can closely control the thickness of a film simply by determining the appropriate concentration of particles and number of nanoparticle layers.

Lvov et al. also observed that SiO_2, an anionic nanoparticle, could not adsorb onto the anionic PSS bonding agent. Additionally, SiO_2 could not be adsorbed onto another SiO_2 layer without first depositing a cationic bonding agent between the two layers. These results confirmed Iler's observation that the alternation of cationic and anionic layers is necessary for successful self-assembly of molecular monolayers. The team also investigated a claim published in [39], which stated that the rinsing of the sample between deposition steps would result in the desorption of some of the colloidal particles. However, Lvov et al. found that their results were reproducible regardless of the number of rinsing steps used. This result indicates that little to no desorption occurs during rinsing.

[3] They used a deposition time of around 15 minutes.

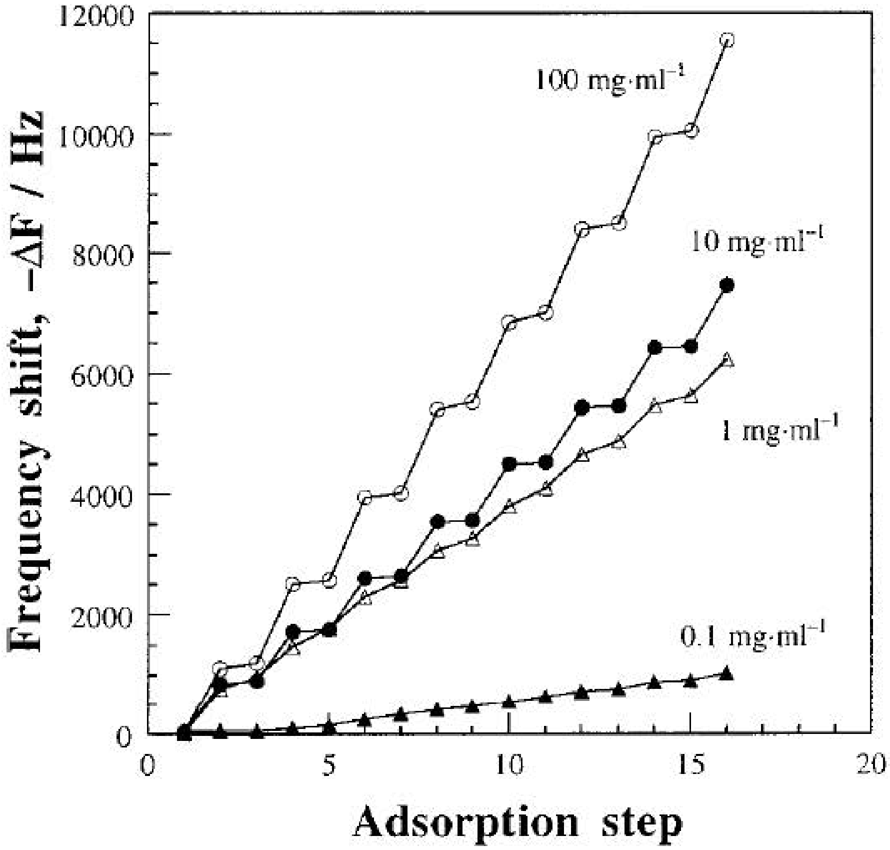

Figure 1.1: Frequency shift in Lvov et al.'s quartz crystal microbalance per adsorption step for several different particle suspension concentrations. The thickness of the adsorbed film varies directly with the frequency shift. Notice that deposition of the bonding agent (odd adsorption steps) produces minimal changes in frequency, while particle deposition (even adsorption steps) increases the frequency at a consistent rate. (Reproduced from [27].)

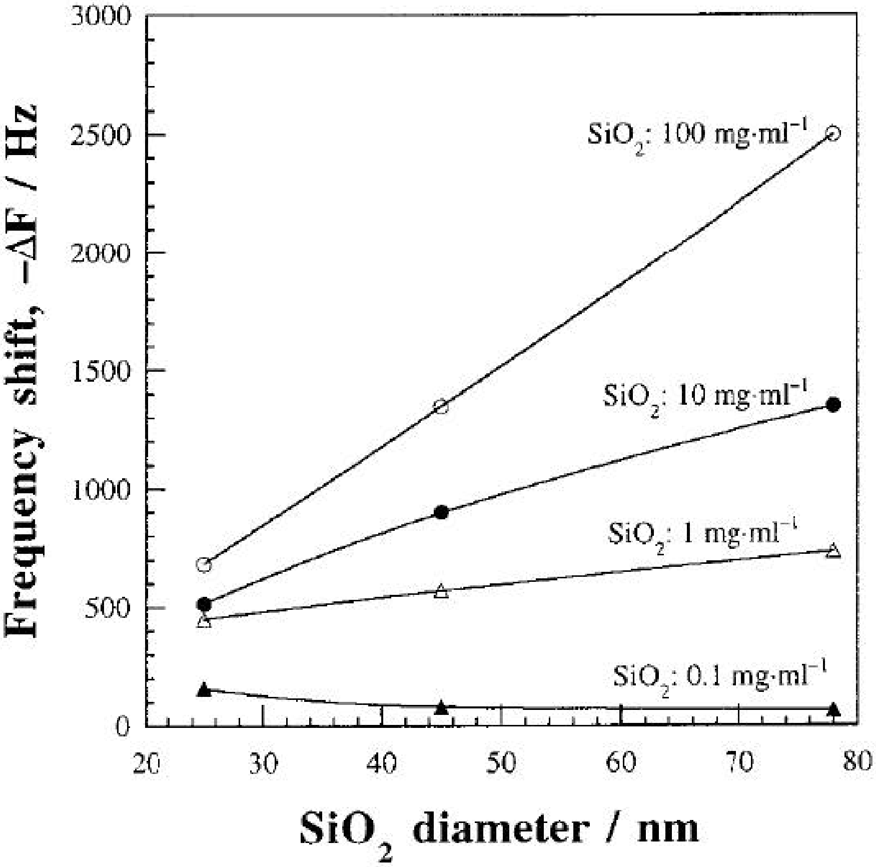

Figure 1.2: Frequency shift in Lvov et al.'s quartz crystal microbalance versus particle size for several different particle suspension concentrations. Notice that the frequency shift, and, consequently, the film thickness, depends upon nanoparticle concentration but not nanoparticle size. (Reproduced from [27].)

1.3 Applications of Self-Assembly

Self-assembling nanoparticles play an important role in a variety of engineering applications. Perhaps the most direct application of self-assembled bilayers is the production of anti-reflective coatings [40]. In an anti-reflective coating, light is incident upon two surfaces. The first is the surface of a thin-film (the coating), and the second is the surface of the coated substrate. At each point of incidence, some of the light refracts while the remainder reflects. We depict this arrangement in Figure 1.3. An anti-reflective coating exploits the physical geometry of the thin film and the difference between the indices of refraction of the film and substrate to cause one of the reflected light rays of experience a π-phase shift with respect to the other.[4] This produces destructive interference between the two reflected beams, which, in turn, prevents light from reflecting off of the coated material. Therefore, all of the incident light passes through the coated material. Anti-

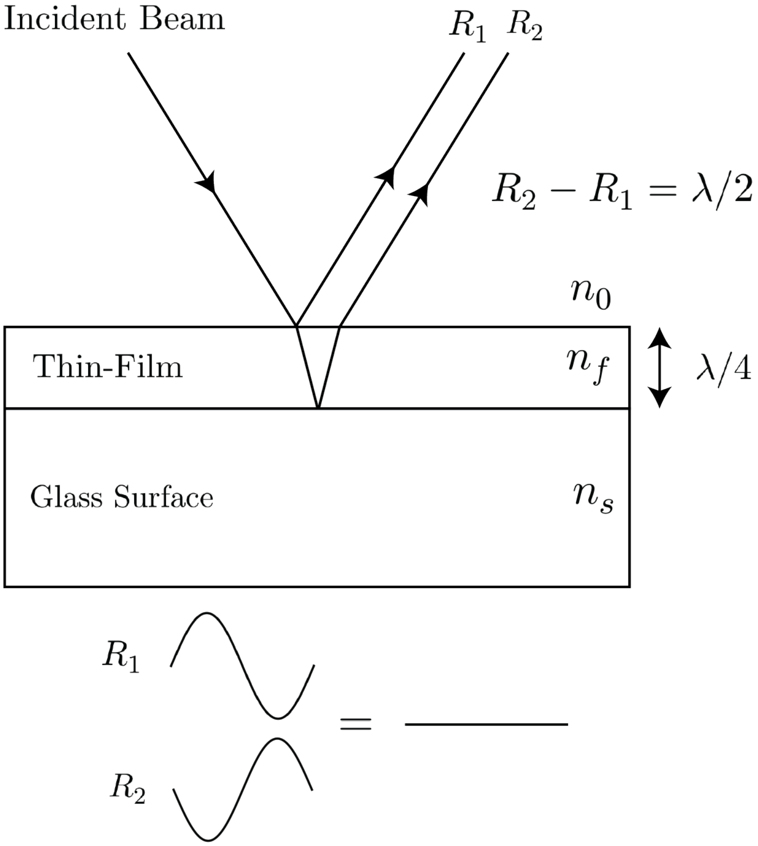

Figure 1.3: The ray diagram for an anti-reflective coating. When the width of the thin film is an integer multiple of $\lambda/4$, the path length difference between the beam that reflects off the thin-film (R_1) and the beam that reflects off the glass surface (R_2) is an integer multiple of $\lambda/2$. This produces a π-phase shift, which results in destructive interference. Thus, no light reflects, and all the electromagnetic energy passes through the glass.

[4] A π-phase shift corresponds to an optical path length difference which is an integer multiple of $\lambda/2$, as we show in Figure 1.3.

reflective coatings are often applied to eyeglasses, increasing the amount of light that can pass through the glass and reducing glare. As described in [36, 37, 38], bilayers of silica and cationic macromolecules (like PDDA) can be used to produce anti-reflective coatings, with the number of bilayers applied and layer separation affecting the reflectance.

Another application of nanoparticle assembly is drug delivery. As explained in [41], nanoparticles are often used to protect drug molecules until they reach a particular part of the body, creating a type of targeted drug delivery. This process is often completed using dendrimers, or tree-like assemblies of nanoparticles. Figure 1.4 is a geometric diagram of a dendrimer. Notice how the nanoparticles (white circles) form branches, which enclose cavities called *dendric boxes* that hold the drug molecules (gray octagons). When the dendrimer reaches the targeted area, the branches open, destroying the dendric boxes and releasing the drug molecules. [42] provides a description of one investigation which uses self-assembling particles in this way.

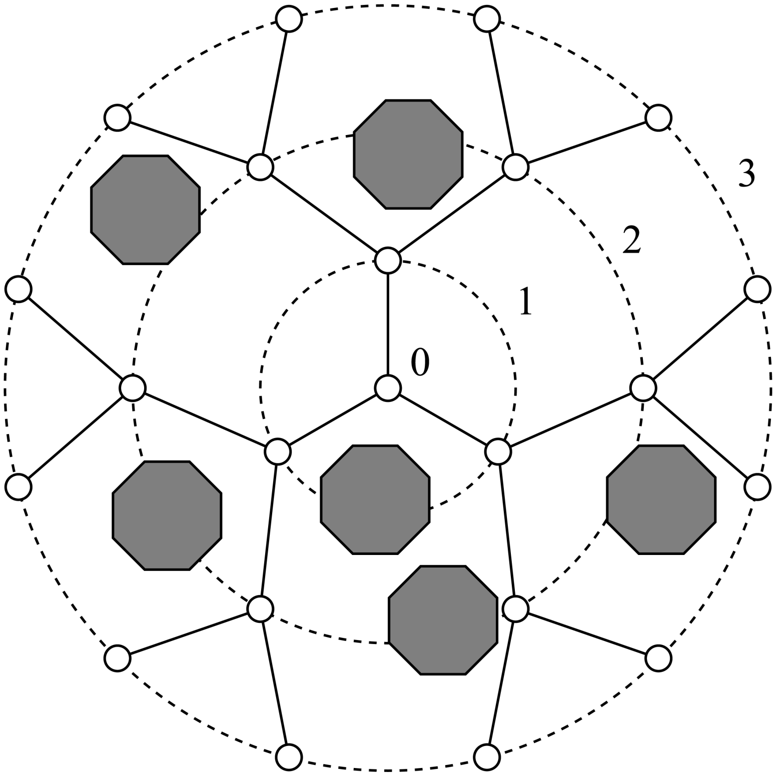

Figure 1.4: A geometric diagram of a dendrimer used for drug encapsulation. The white circles are nanoparticles which self-assemble in a branching pattern. The solid black lines indicate weak bonds between nanoparticles. The gray octagons are drug molecules. The drug molecules sit in openings between the branches known as dendric boxes. When the dendrimer reaches the intended target, the branches open, releasing the drug molecules. (Reproduced from [41].)

In electrical engineering, nanoparticle self-assembly serves as an important tool in the development of nano-scale circuitry. [43] describes the use of self-assembly techniques to arrange electrochemically synthesized nanowires into viable electrical circuits. Nano-scale circuits, whether constructed using more traditional top-down approaches or via bottom-up approaches like stochastic assembly, have made possible massive advances in computing capabilities. For this reason, we anticipate further work in this area.

Finally, nanoparticle self-assembly also facilitates more controlled nano-construction techniques like nanoprinting. As explained in [44], nanoprinting via self-assembly allows scientists to easily copy complicated nanostructures. In many cases, these techniques permit construction times that are far less than required by more traditional techniques, including electron beam lithography and scanning probe lithography. Nanoprinting's ability to revolutionize the reproducibility of more complicated structures makes this a particularly active area of research.

1.4 Imaging Techniques

1.4.1 Scanning Electron Microscopy (SEM)

Scanning electron microscopy [2], the primary technique used throughout our investigation for the imaging of nanoparticle monolayers, is a class of experimental methods which use the wave-like properties of electrons to produce an image, just as the wave-like properties of photons are used to produce an image in an optical microscope. SEM was pioneered by Ernst Ruska [45], who in 1931 demonstrated that the image of a grid could be magnified if the grid were placed after a converging electron beam's focal point. Ruska's work noted that the magnification m produced by his electron beam behaved according to the rules of geometric optics, which state that

$$m = -\frac{s'}{s}, \tag{1.3}$$

where s' is the image distance (the distance between the lens and the image) and s is the object distance (the distance between the sample and the lens) [2]. Optical microscopes are limited by the finite wavelength of a photon (i.e., the cannot image any object that is smaller than the wavelength of light used to illuminate it). This same limitation exists for electrons, which according to de Broglie, also have an associated wavelength λ. However, de Broglie's formula

$$\lambda = \frac{h}{mv}, \tag{1.4}$$

where λ is the wavelength of the particle, h is Planck's constant, m is the particle mass, and v is the magnitude of the particle velocity, predicts that electrons (moving at speeds typical of those that can be produced in an electron microscope) possess wavelengths anywhere from 0.08 Å to 0.03 Å. With most atoms having radii on the order of a few angstroms, these values indicate that SEM techniques can, in principle, achieve atomic resolution. For imaging on the nanoscale, SEM is more than sufficient.

Most SEM imaging is achieved according to the following method [2]. First, an electron beam is produced using either thermionic emission or field emission. In thermionic emission (the method used by our SEM), a filament is heated, increasing the energy of internal electrons until they overcome an energy barrier, denoted by a work function ϕ (see Figure 1.5) [12]. Electrons that have overcome this energy barrier can then be collected into an electron beam. The current density of the produced electron beam is given by Richardson's law

$$J = \frac{4\pi me}{h^3}(k_BT)^2\exp\left(-\frac{\phi}{k_BT}\right), \tag{1.5}$$

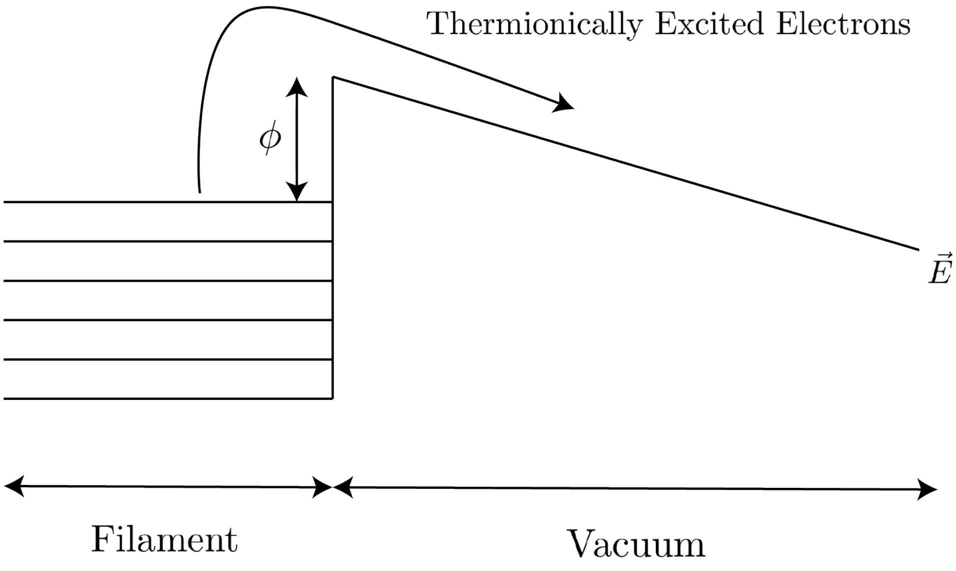

Figure 1.5: Electron beam production via thermionic emission. When the filament is heated, electrons are excited over the energy barrier, which is indicated by the work function ϕ.

where m is the mass of an electron, e is the fundamental charge, h is Planck's constant, k_B is Boltzmann's constant, T is the temperature of the filament, and ϕ is the work function of the filament [2]. Alternatively, in field emission, a strong external electric field is used to encourage the electrons in a source material to tunnel through the energy barrier (see Figure 1.6) [12]. For this

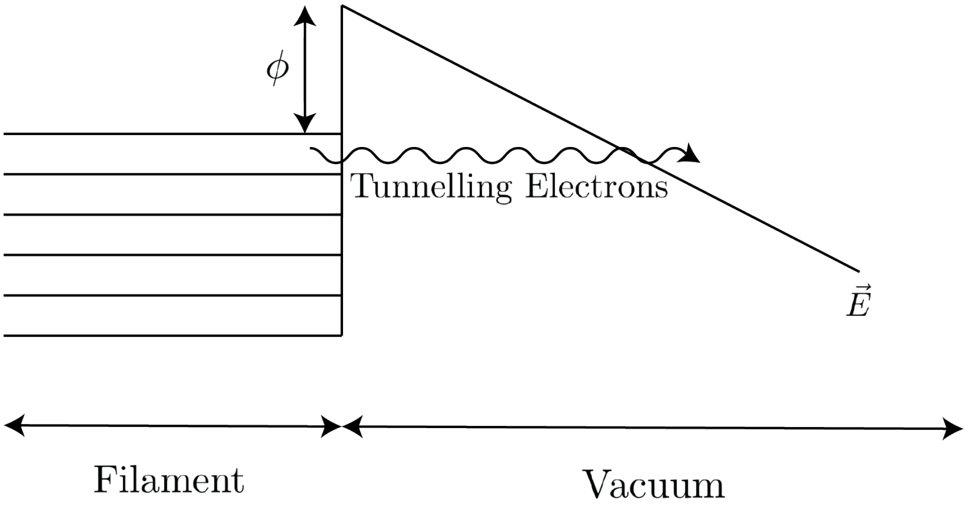

Figure 1.6: Electron beam production via field emission. Under the influence of a strong electric field, electrons in the filament are encouraged to tunnel through the energy barrier, indicated by the work function ϕ.

method, the current density of the produced electron beam is proportional to the tunneling rate, given by the Fowler-Nordheim equation

$$J \propto E^2 \exp\left(-\frac{4\sqrt{2m}}{3e\hbar}\frac{\phi^{3/2}}{E} \right), \tag{1.6}$$

where E is the magnitude of the electric field used to induce tunneling, m is the mass of the electron, e is the fundamental charge, $\hbar$ is the reduced form of Planck's constant, and ϕ is the work function of the source material [2]. While thermionic emission is more typical, field emission is capable of higher resolution due to a higher consistency in electron wavelength. Furthermore, the aberrations in electron lenses have less of a negative effect on an electron beam produced via field emission than a beam produced via thermionic emission.

After a beam of electrons has been produced, a series of magnetic lenses are used to focus the beam on the surface of the sample. The path of an electron deflected by electric and magnetic fields is given by

$$m\frac{d^2\boldsymbol{r}}{dt^2} = -e\boldsymbol{v} \times \boldsymbol{B} - e\boldsymbol{E}, \tag{1.7}$$

where m is the mass of the electron, $\boldsymbol{r}$ is the electron's position vector, e is the fundamental charge, $\boldsymbol{v}$ is the velocity of the electron, $\boldsymbol{B}$ is the magnetic field vector, and $\boldsymbol{E}$ is the electric field vector. This equation predicts a motion similar to that depicted in Figure 1.7, which shows a series of two magnetic lenses being used to focus a diverging electron beam originating at point O onto an image plane at point I. The vector product between $\boldsymbol{v}$ and $\boldsymbol{B}$ forces the electrons to spiral around the magnetic field [2].

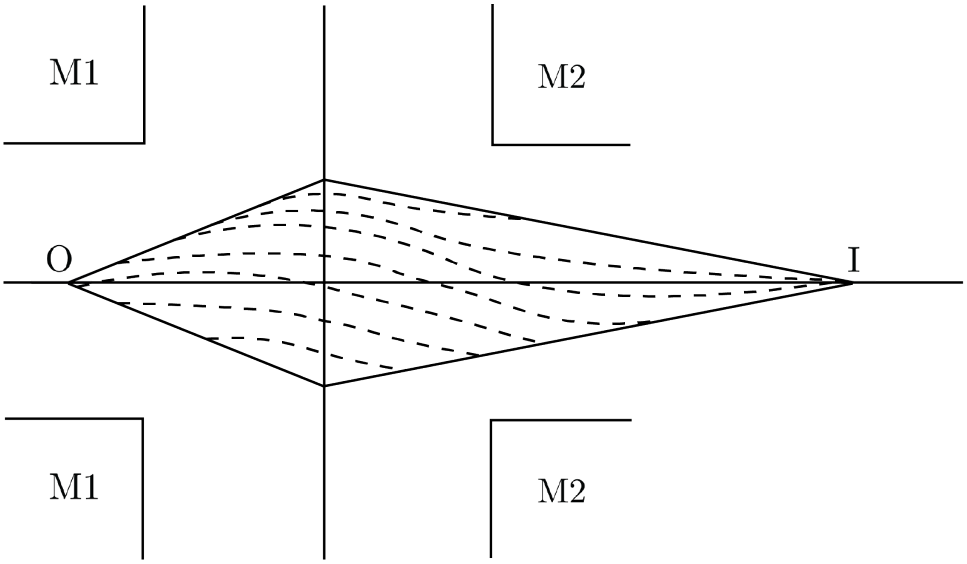

Figure 1.7: Electron beam focusing via magnetic lensing. The two electromagnets, M1 and M2, cause the beam originating at point O to focus at point I.

Once the beam has been focused into a small spot on the sample, magnetic deflection of the beam is used to raster across the sample's surface. The image is then produced in one of several ways, all of which depend upon the interaction between the electron beam and the sample at their point of contact. For especially thin samples, the electrons striking the surface are collected directly using a sensor placed below the sample. In this scenario, often called scanning transmission electron microscopy (STEM), the transmitted current is used to produce the image. Thicker samples often rely upon the collection of elastically backscattered electrons, or even inelastically scattered electrons. SEMs that can achieve especially high electron beam energies can even excite and detect X-ray emissions from the electrons in the sample [2].

1.4.2 Atomic Force Microscopy (AFM)

Another emerging method for the imaging and manipulation of nanoscale structures is atomic force microscopy (AFM). Developed by Binnig, Quate, and Gerber [13] in 1986 to address several limitations of scanning tunneling microscopy (another imaging method), AFM measures the deflections of a cantilever tip to produce a reliable image of a surface. Advantages of this method include the ability to directly manipulate atoms or molecules on a sample as well as the option of producing three dimensional images and sample height profiles. AFM techniques are also commonly used to measure the electrical properties of samples.

As depicted in Figure 1.8, AFM imaging begins with a laser source, which reflects off of a cantilever arm and is detected by a two segment photodiode. As the cantilever arm rasters across the sample, atoms within the sample deflect the tip, causing the entire cantilever arm to vibrate at a frequency given by

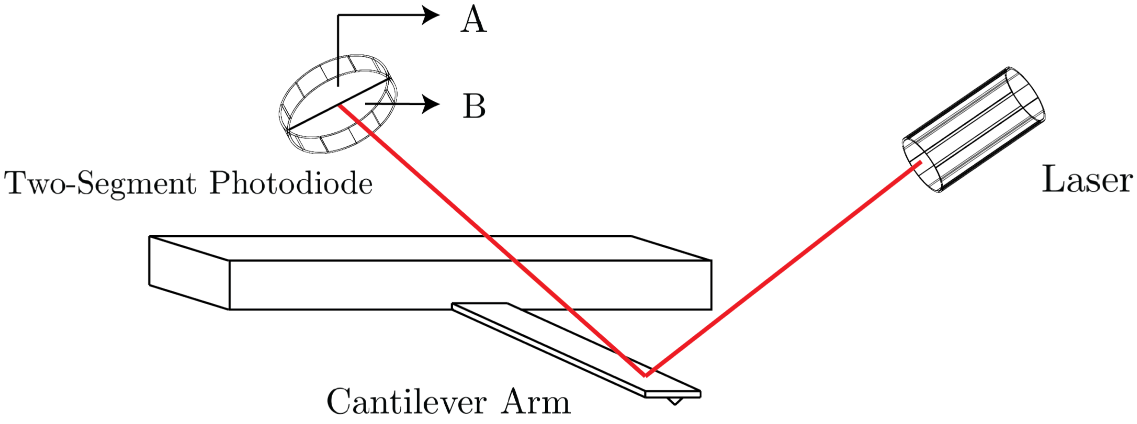

Figure 1.8: The AFM detection process. A laser beam bounces off of a cantilever arm, which is deflected by atoms on the surface of the sample. A two-segment photodiode collects the reflected light, using the quotient of the difference and sum of the two photodiode currents to calculate the arm's deflection. The AFM combines deflection information obtained from across the sample to produce an image.

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{k}{m}}, \tag{1.8}$$

where k is the effective spring constant of the arm and m is its mass. The image is ultimately produced using information about the deflection of the cantilever arm (δz), which is proportional to the quotient of the difference and sum of the two photodiode currents (i_A and i_B):

$$\delta z \propto \frac{i_A - i_B}{i_A + i_B}. \tag{1.9}$$

AFM can be performed in several different modes, including contact mode, in which the tip is allowed to make direct contact with the sample, non-contact mode, in which the tip is held a set distance above the sample and deflected by electrostatic repulsion, and vibrating mode, in which the tip is allowed to vibrate on top of the sample [2].

Chapter 2

Self-Assembly under No External Field (ISAM)

2.1 ISAM Overview

Building on the work of Iler and Lvov, our models, simulations, and experiments concern thin films constructed from uniform layers of nanoparticles. To produce these layers, we use the ionic self-assembly of monolayers process (ISAM). Figure 2.1 details the ISAM process when used to adhere negatively charged particles to a flat glass surface such as a microscope slide. Due to the presence of silica in glass, the surface of any standard glass slide immersed in water has an innate negative

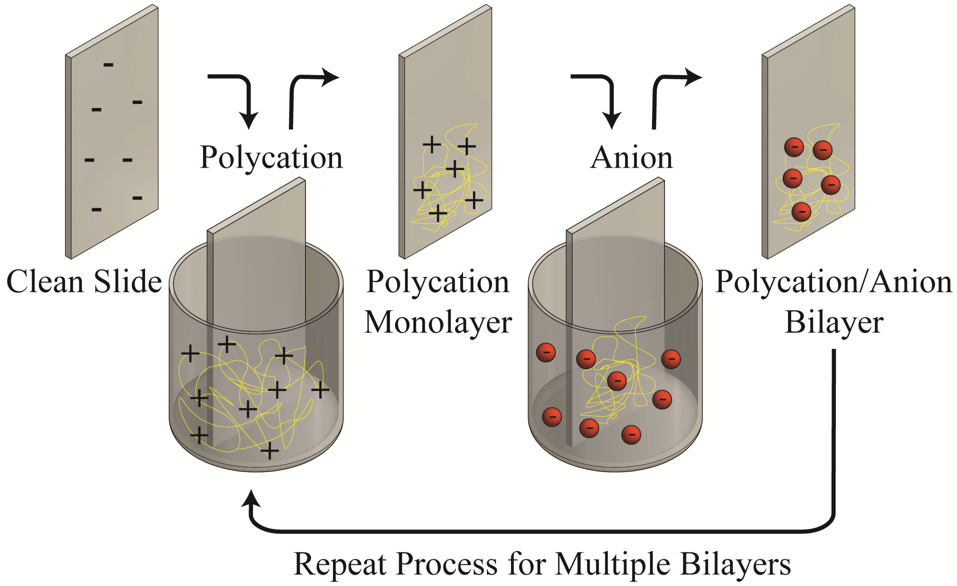

Figure 2.1: The standard ISAM process for negatively charged nanoparticles, in which a clean slide with a slight negative charge becomes positively charged due to exposure to a polycationic suspension. The positively charged slide then enters the nanoparticle suspension, where nanoparticles self-assemble on its surface according to electrostatic forces. (Reproduced from [41, 46].)

charge [47]. This is indicated by the negative signs on the surface of the Clean Slide in Figure 2.1. We first dip the slide in a suspension of poly(diallyldimethylammonium chloride) (PDDA), which is a standard polycation. Its positively charged molecules easily adhere to the surface of the negatively charged slide, forming a single monolayer of positively charged ions. This is indicated by the positive signs on the Polycation Monolayer Slide shown in Figure 2.1. After producing the polycation monolayer, we then dip the slide in a suspension of silicon dioxide (SiO_2) nanoparticles, which is anionic. Thus, the electrostatic forces present between the cationic PDDA and the anionic SiO_2 cause the SiO_2 to adhere to the surface of the slide. We now have a single bilayer formed from individual monolayers of PDDA and SiO_2, as indicated by the Polycation/Anion Bilayer Slide in Figure 2.1. By repeating the ISAM process, we can produce thin films with any number of bilayers.

2.2 CSAE Modeling Techniques and Results

Physicists and engineers naturally desire a means of predicting the properties of a film produced via ISAM. For this purpose, a variety of techniques have been developed using principles from statistical physics to model the nanoparticle assembly process. By connecting the assembly process to emergent properties such as particle coverage density, it is possible to accurately predict the optical properties of a thin film produced via ISAM.

One of the primary modeling techniques for nanoparticle assembly is the use of a cooperative sequential adsorption with evaporation (CSAE) model [41, 48]. CSAE models imagine the assembly surface as a grid, as shown in Figure 2.2. Each location on the grid is called a site and is indicated mathematically by a site number i. Each site can exist in one of two states, indicated by a state value n_i. Sites that contain particles (colored black in Figure 2.2) are considered occupied and are denoted mathematically by $n_i = 1$. Sites that do not contain particles (colored white in Figure 2.2) are considered unoccupied and are denoted mathematically by $n_i = 0$. Sites transition between

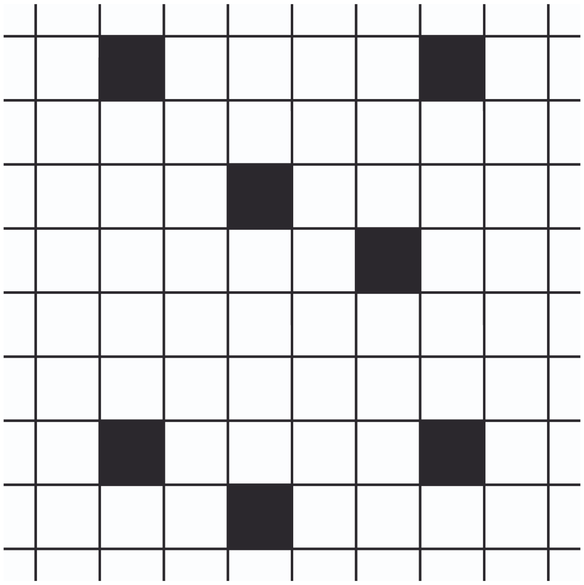

Figure 2.2: A square grid of the type used in CSAE modeling. Black squares indicate occupied sites ($n_i = 1$), while white squares indicate unoccupied sites ($n_i = 0$). (Reproduced from [41].)

states according to a transition rate $c(n_i \to (1 - n_i))$. The rules of the transition rate depend upon the particular CSAE model used. CSAE models differ in their complexity, which corresponds to the amount of physical detail they encompass. Our work considers two different CSAE models: the total lattice model (CSAE-TL) and the nearest neighbors model (CSAE-NN). We consider the construction and use of each model in detail in the following sections.

2.2.1 CSAE-TL

The transition rate for the CSAE-TL model [41] is given in Equation (2.1):

$$c_{TL}(n_i \to (1 - n_i)) = n_i\gamma + \mu(1 - n_i)\left(1 - \frac{\sum_{i=1}^{n} n_i}{N}\right). \tag{2.1}$$

When a site is occupied (i.e., $n_i = 1$), the first term, $n_i\gamma$, is active. This term is called the evaporation term because it allows the particle to detach from the surface of the slide (a transition $c_{TL}(1 \to 0)$) with a probability $\gamma \in [0, 1]$. For this reason, γ is called the evaporation coefficient. When a site is unoccupied (i.e., $n_i = 0$), the second term, $\mu(1 - n_i)\left(1 - \frac{\sum_{i=1}^{n} n_i}{N}\right)$, is active. This term is called the deposition term because it allows a particle to attach to the surface of the slide at the unoccupied site (a transition $c_{TL}(0 \to 1)$). The probability of deposition relies upon the deposition coefficient $\mu \in [0, 1]$. Unlike the evaporation coefficient γ, μ changes as the number of particles on the slide changes. This behavior is important because charged nanoparticles experience electrostatic repulsion when in proximity to one another. The CSAE-TL model considers the previously deposited particles as one single charge screen (see Figure 2.3). Thus, the probability

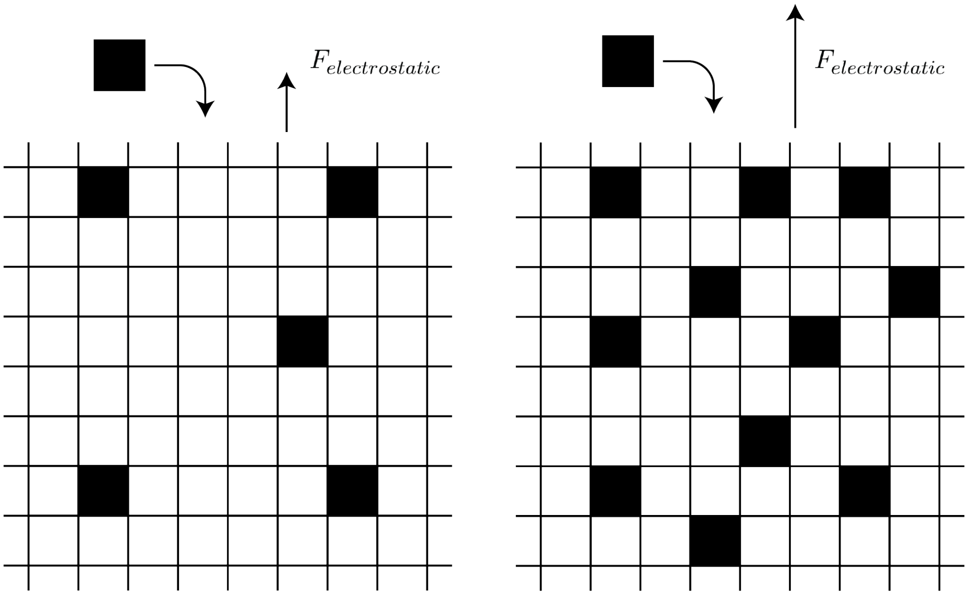

Figure 2.3: The total-lattice approach to CSAE. The left grid has a low occupation number. The resulting electrostatic force ($F_{electrostatic}$) is weak, allowing the particle to easily deposit onto the grid. The right grid has a high occupation number. The resulting electrostatic force ($F_{electrostatic}$) is strong, making particle deposition more difficult.

of deposition, which begins at μ when no particles have been deposited, decreases as the fraction of occupied sites $\frac{\sum_{i=1}^{n} n_i}{N}$ increases. (N indicates the total number of sites on the lattice.)

While the CSAE-TL model fails to account for the distribution of charge on the surface of the slide, it produces an analytically solvable equation for steady state particle coverage density. This makes the CSAE-TL modeling method particularly useful in situations where exact charge distribution detail is not necessary or when numerical solution methods are not available.

Converting the CSAE-TL transition rate into an equation for coverage density requires the use of mean field theory, a collection of approximative methods from statistical physics (see Appendix A for more detail). By assuming that edge effects are negligible in the interior of the slide and that particles tend to distribute themselves evenly across the slide, we reason that particle coverage density, ρ, does not depend upon location on the slide. Thus, mean field theory's assumption that each site will have approximately the same number of occupied neighbor sites and feel the same effect is valid. This allows us to replace the local 'field' felt by each particle with an overall mean field that is felt by the entire lattice.

Mathematically, we use the mean field approximation as follows. We begin with the partial differential equation

$$\frac{\partial \langle n_i \rangle}{\partial t} = -\gamma \langle n_i \rangle + \mu \left\langle (1 - n_i) \left(1 - \frac{\sum_{i=1}^{n} n_i}{N} \right) \right\rangle \tag{2.2}$$

where $\langle n_i \rangle$ indicates the mean individual site occupation. Because we assert that neighboring sites are uncorrelated, we can assume that the ensemble average of nearest neighbor correlations is approximated by the product of the mean individual site occupations. Mathematically, this means that

$$\langle n_i n_j \rangle = \langle n_i \rangle \langle n_j \rangle, \tag{2.3}$$

where n_i and n_j are any two neighboring sites. This equality reduces our equation as follows:

$$\frac{\partial \langle n_i \rangle}{\partial t} = -\gamma \langle n_i \rangle + \mu (1 - \langle n_i \rangle) \left(1 - \frac{\sum_{i=1}^{n} \langle n_i \rangle}{N} \right). \tag{2.4}$$

Technically, each site n_i has a different rate equation like the one expressed in Equation (2.2). However, if we assume every site is relatively the same, we can reason that

$$\langle n_i \rangle = \langle n \rangle. \tag{2.5}$$

This assumption removes the site-specific nature of our rate equations, allowing us to apply the same equation for all sites. Finally, we define particle density ρ as

$$\rho = \frac{\sum \langle n_i \rangle}{N}. \tag{2.6}$$

This gives us the following partial differential equation for particle coverage density:

$$\frac{\partial \rho}{\partial t} = -\gamma \rho + \mu (1 - \rho)^2. \tag{2.7}$$

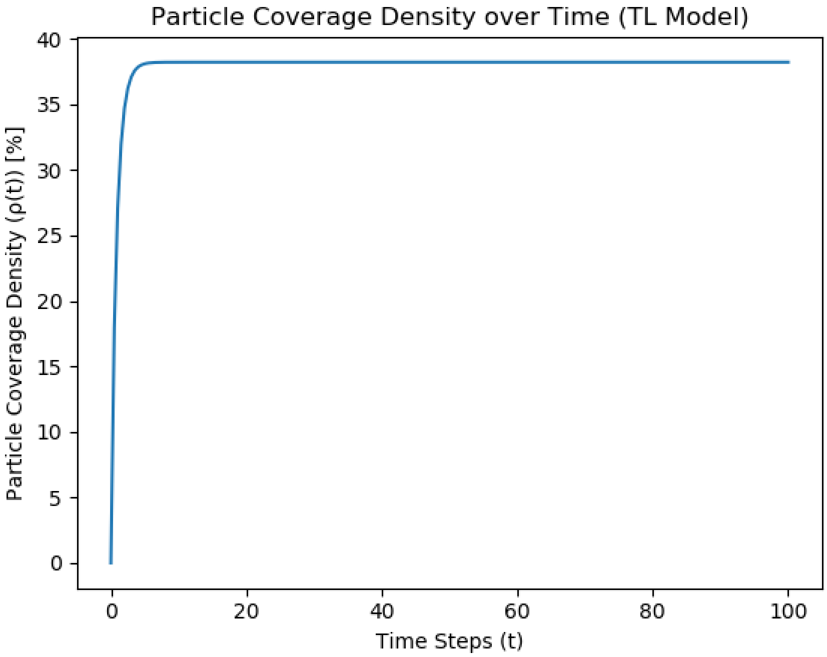

Figure 2.4: The numerical solution to our CSAE-TL model (Equation (2.7)) when $\gamma = \mu = 0.5$. Notice that the particle coverage density ρ increases rapidly until reaching a steady state of $\rho_s \approx 38.2\%$.

Equation (2.7) suggests that ρ changes over time. Using the `odeint` function from Python's SciPy package (see Appendix B.1 for a description of `odeint` and Appendix C.1 for a detailed look at our program), we can produce plots depicting the coverage density over time. Figure 2.4 is an example of such a plot. In this case we use $\gamma = \mu = 0.5$. Notice how the particle coverage density (which we have denoted as a percentage of slide area) rapidly increases until a steady state is achieved. The particle coverage density at steady state ρ_s can be easily calculated without the use of numerical solving methods like `odeint` by simply setting $\frac{\partial \rho}{\partial t} = 0$ in Equation (2.7) and solving for ρ:

$$\rho_s = \frac{2\mu \pm \sqrt{\gamma(4\mu + \gamma)} + \gamma}{2\mu}. \tag{2.8}$$

When $\gamma = \mu = 0.5$, we calculate $\rho_s \approx 38.2\%$, which matches the steady state achieved in Figure 2.4.[1]

Figures 2.5 and 2.6 show how ρ_s varies as γ and μ change, respectively. From these plots, we make three primary observations. First, the steady state coverage density ρ_s decreases steadily as the evaporation coefficient γ increases. This mathematical behavior mirrors what we would expect physically: a higher chance of particle detachment corresponds to a lower steady state coverage density. Second, ρ_s increases as the deposition coefficient μ increases. This result also makes sense physically, as a higher likelihood of particle deposition should correspond to a higher steady state

[1] The second solution, $\rho_s \approx 262\%$ is discarded because it describes a non-physical scenario.

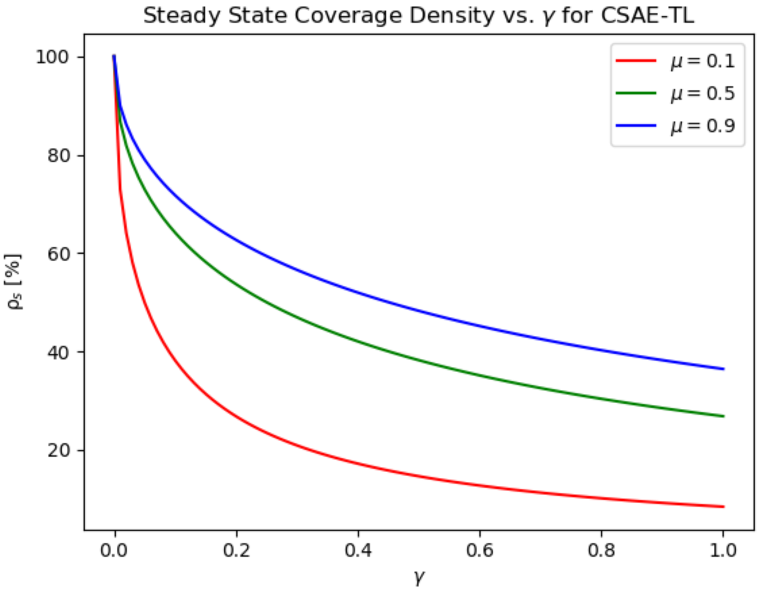

Figure 2.5: The γ response of steady state particle coverage density calculated using our CSAE-TL model. Notice that a higher γ value corresponds to a lower ρ_s value. Also notice that changing μ alters the value of ρ_s but not the shape of the response curve.

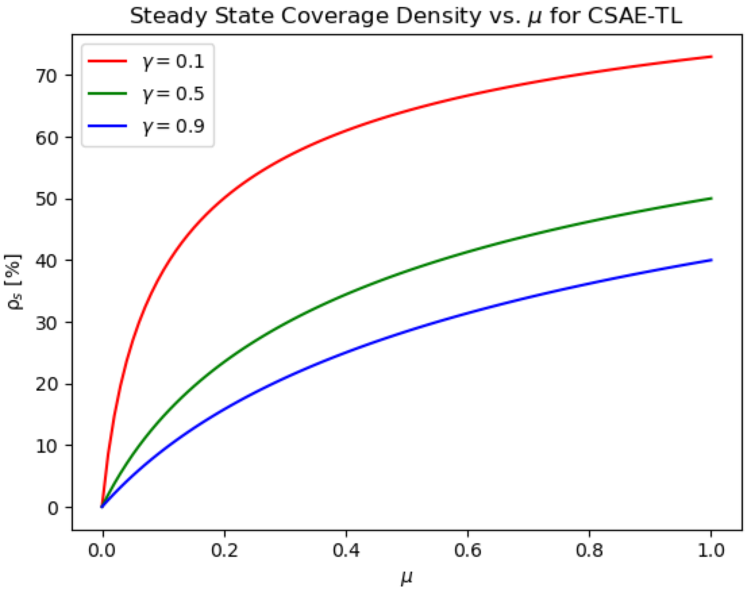

Figure 2.6: The μ response of steady state particle coverage density calculated using our CSAE-TL model. Notice that a higher μ value corresponds to a higher ρ_s value. Also notice that changing γ alters the value of ρ_s but not the shape of the response curve.

coverage density. Third, we note that the slopes of the curves in both Figure 2.5 and Figure 2.6 are steeper for lower values of γ and μ, respectively. This non-linear behavior indicates that small changes in γ or μ have a greater impact when these values are lower on the lower end of their domain than when they are on the higher end.

2.2.2 CSAE-NN

The transition rate for the CSAE-NN[2] model [41, 48] is given in Equation (2.9):

$$c_{NN}(n_i \to (1-n_i)) = n_i\gamma + (1-n_i)\alpha\beta^{\eta}. \tag{2.9}$$

When a site is occupied (i.e., $n_i = 1$), the first term, $n_i\gamma$, is active. Like in the CSAE-TL model, this term is called the evaporation term because it allows the particle to detach from the surface of the slide (a transition $c_{TL}(1 \to 0)$) with a probability $\gamma \in [0,1]$. For this reason, γ is once again called the evaporation coefficient. When a site is unoccupied (i.e., $n_i = 0$), the second term, $(1-n_i)\alpha\beta^{\eta}$, is active. This deposition term allows a particle to attach to the surface of the slide at the unoccupied site (a transition $c_{TL}(0 \to 1)$); however, it functions quite differently from the deposition term found in the CSAE-TL model. The probability of deposition is now governed by three different parameters: α, β, and η. $\alpha \in [0,1]$ describes the probability of deposition when no nanoparticles are present on the slide surface. Like the decrease from μ in the CSAE-TL model, the probability of deposition will decrease from α as more particles occupy the slide. However, the CSAE-NN model takes a more detailed approach, decreasing the deposition probability from α according to the number of deposited particles neighboring the site in question (hence the nearest neighbors model name). This approach ensures that local variations in the electrostatic force created by previously deposited nanoparticles are taken into account (see Figure 2.7). Two parameters are needed to describe this variable deposition probability. $\beta \geq 0$ describes the strength of the electrostatic force which causes individual SiO_2 particles to repel one another. $\eta = \sum_{j \in NN} n_j$ quantifies the number of sites neighboring site n_i. As the number of occupied neighboring sites increases, the strength of electrostatic repulsion increases exponentially. Thus, the CSAE-NN model describes a scenario in which particle deposition becomes increasingly difficult as a particular region of the slide becomes more occupied with nanoparticles

Like with the CSAE-TL model, the transition rate for the CSAE-NN model can be converted into a partial differential equation describing the change in particle coverage density over time. Once again, the mean field approximation is integral to this conversion. We begin with the following rate equation:

$$\frac{\partial \langle n_i \rangle}{\partial t} = -\gamma\langle n_i \rangle + \langle (1-n_i)\alpha\beta^{\eta} \rangle. \tag{2.10}$$

Because the application of mean field theory allows us to approximate the correlations between neighboring sites as

$$\langle n_i n_j \rangle = \langle n_i \rangle \langle n_j \rangle, \tag{2.11}$$

Equation (2.10) reduces to

$$\frac{\partial \langle n_i \rangle}{\partial t} = -\gamma\langle n_i \rangle + (1-\langle n_i \rangle)\alpha\langle \beta^{\eta} \rangle. \tag{2.12}$$

[2] As a reminder to the reader, CSAE-NN stands for cooperative sequential adsorption with evaporation using the nearest neighbors approach.

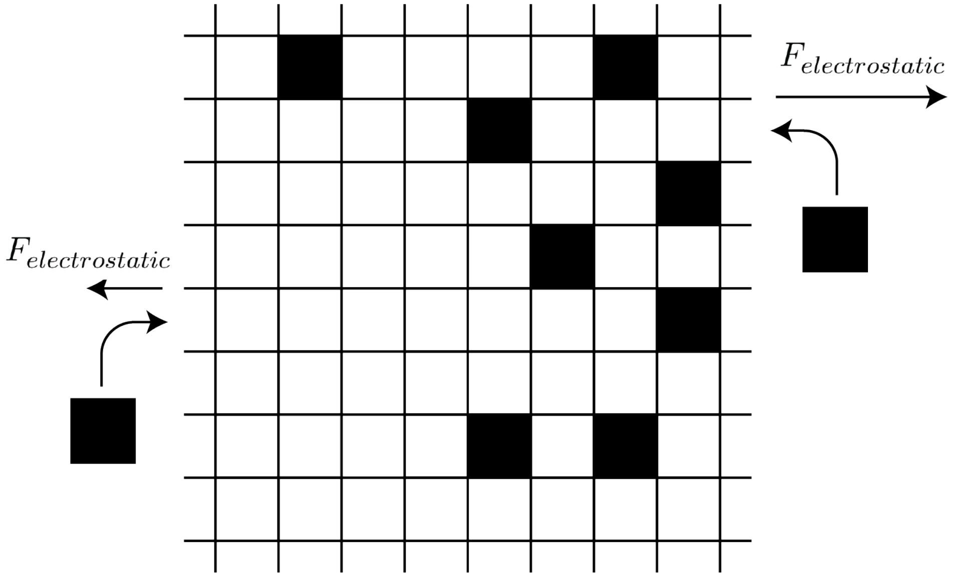

Figure 2.7: The nearest neighbor approach to CSAE. The depositing particle on the left experiences a weak electrostatic force ($F_{electrostatic}$) since few particles are in that region of the grid. Thus, it may deposit easily. The depositing particle on the right experience a strong electrostatic force ($F_{electrostatic}$) since many particles already occupy that region of the grid. This makes deposition more difficult.

By performing a Taylor series expansion on $\langle\beta^\eta\rangle$, we can show that, to a first approximation, $\langle\beta^\eta\rangle = \beta^{\langle\eta\rangle}$:

$$\begin{aligned}\langle\beta^\eta\rangle = \langle e^{\eta\ln\beta}\rangle &= \left\langle 1 + \eta\ln\beta + \frac{1}{2}(\eta\ln\beta)^2 + \cdots \right\rangle &(2.13)\\ &= 1 + \langle\eta\rangle\ln\beta + \frac{1}{2}\langle\eta^2\rangle\ln^2\beta + \cdots &(2.14)\\ &= 1 + \langle\eta\rangle\ln\beta + \frac{1}{2}\langle\eta\rangle^2\ln^2\beta + \cdots &(2.15)\\ &= e^{\langle\eta\rangle\ln\beta} &(2.16)\\ &= \beta^{\langle\eta\rangle}. &(2.17)\end{aligned}$$

Equation (2.12) now becomes

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\gamma\langle n_i\rangle + (1 - \langle n_i\rangle)\alpha\beta^{\langle\eta\rangle}. \tag{2.18}$$

Like with the CSAE-TL model, we eliminate the need for site-specific equations by assuming that $\langle n_i\rangle = \langle n\rangle$. For the CSAE-NN model, this also means that $\langle\eta\rangle = z\langle n\rangle$, where z is the mean number of the nearest neighbors for each site.[3] Defining the particle coverage density ρ once again as

$$\rho = \frac{\sum\langle n_i\rangle}{N}, \tag{2.19}$$

[3] For a square lattice, $z = 4$.

we arrive at

$$\frac{\partial \rho}{\partial t} = -\gamma\rho + (1-\rho)\alpha\beta^{z\rho}. \tag{2.20}$$

Using `odeint`, we can produce plots describing the numerical solution to Equation (2.20). Figure 2.8 is an example of such a plot, produced when $\alpha = \gamma = 0.5$, $\beta = 1$, and $z = 4$. Once again, the coverage density increases rapidly until steady state is achieved. Unlike with the CSAE-TL model, however, the CSAE-NN model does not have an analytic solution for the coverage density at steady state ρ_s. This does not mean that an analytic approximation is not possible. To find such an approximation, we begin by setting $\frac{\partial \rho}{\partial t} = 0$ in Equation (2.20):

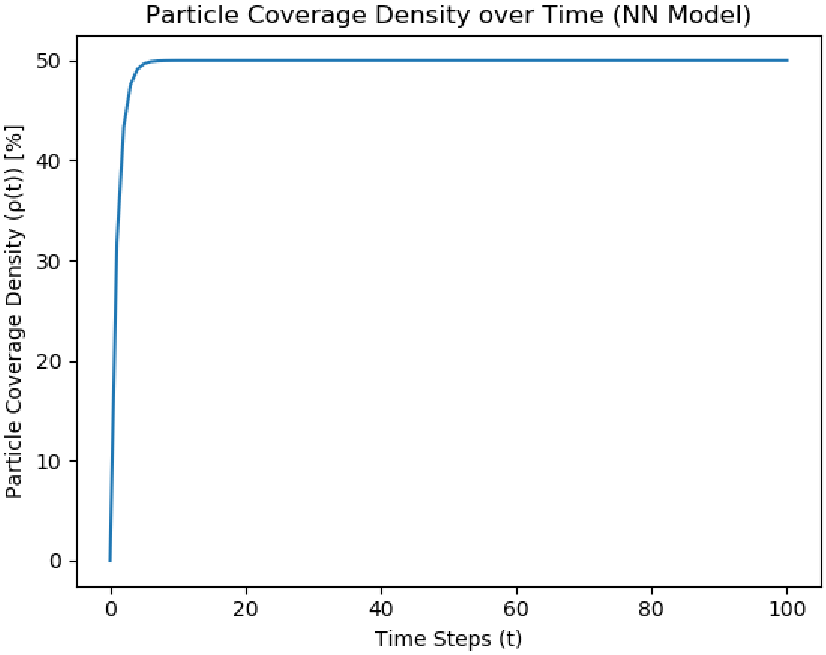

Figure 2.8: The numerical solution to our CSAE-NN model (Equation (2.20)) when $\alpha = \gamma = 0.5$, $\beta = 1$, and $z = 4$. Notice that the particle coverage density ρ increases rapidly until reaching a steady state of $\rho_s \approx 50\%$.

$$0 = -\gamma\rho_s + (1-\rho_s)\alpha\beta^{z\rho_s}. \tag{2.21}$$

This yields the transcendental equation

$$\rho_s = \frac{\alpha\beta^{z\rho_s}}{\gamma + \alpha\beta^{z\rho_s}}. \tag{2.22}$$

This equation can be solved graphically. Analytically, however, we can take the Taylor series expansion about $\beta = 1$:

$$\rho_s = \rho_s(\beta = 1) + (\beta - 1)\frac{\partial \rho_s}{\partial \beta}\bigg|_{\beta=1} + \cdots . \tag{2.23}$$

In our approximation, we keep only the constant and linear terms. If we assume that $\rho_s(\beta = 1) = \frac{\alpha}{\gamma+\alpha}$, we arrive at

$$\rho_s = \frac{\alpha}{\gamma + \alpha} - (1-\beta)\left[4\left(\frac{\alpha}{\gamma+\alpha}\right)^2\left(1 - \frac{\alpha}{\alpha+\gamma}\right)\right]. \tag{2.24}$$

For our example, Equation (2.24) predicts a particle coverage density of $\rho_s \approx 50\%$, which is approximately the same as the steady state predicted by our numerical solution (Figure 2.8).

Figures 2.9 and 2.10 demonstrate how ρ_s changes as γ and β change, respectively. Like with Figures 2.5 and 2.6, these plots show a decrease in particle coverage density ρ_s as the evaporation coefficient γ increases. Conversely, ρ_s increases as the deposition coefficient β increases. These results mirror our physical expectation, as a higher evaporation tendency should reduce the slide's coverage density while a higher deposition tendency should increase coverage density.

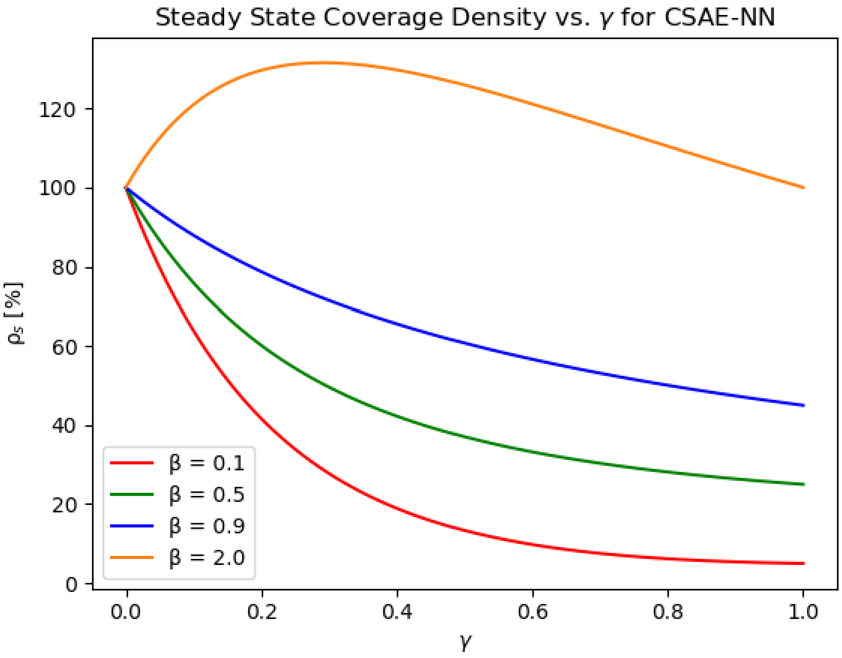

Figure 2.9: The γ response of steady state particle coverage density calculated using our CSAE-NN model. Notice that a higher γ value corresponds to a lower ρ_s value. Also notice that changing β alters the value of ρ_s but not the shape of the response curve for $\beta \leq 1$. When $\beta > 1$ (e.g., the orange curve) the ρ_s response exhibits a maximum. The orange ($\beta = 2.0$) curve represents an over-packing scenario, which occurs when previously deposited particles attract particles that have not yet been deposited.

In Figure 2.9 we include an extra curve plotted when $\beta = 2.0$. This helps to demonstrate the physical correspondence of the β factor and its distinction from α. While α describes the probability of particle deposition before any particles have attached to the slide (identical to μ in the CSAE-TL model), β describes the interactions between particles on the slide and particles attempting to deposit. If $\beta < 1$ (red, green, and blue curves in Figure 2.9), particles present on the slide repel

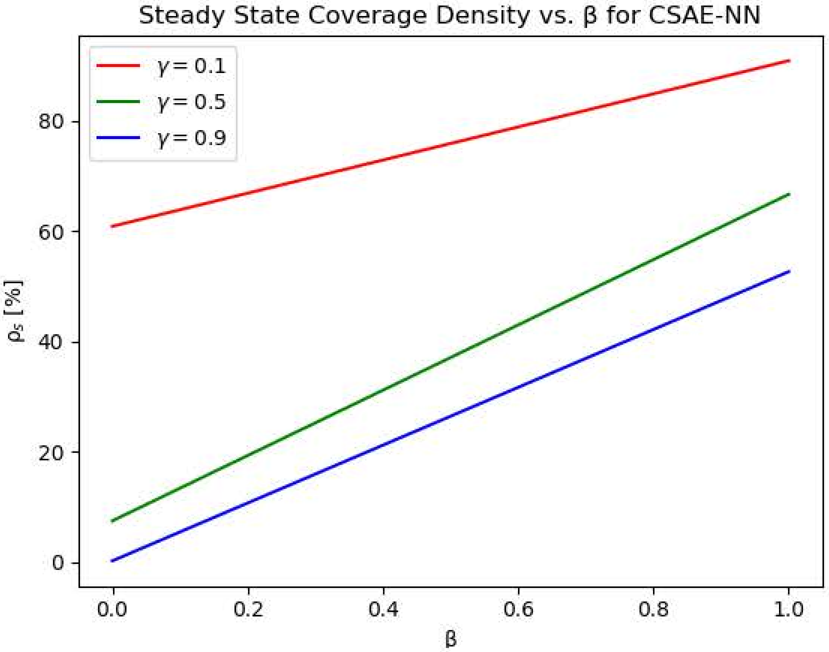

Figure 2.10: The β response of steady state particle coverage density calculated using our CSAE-NN model. Notice that a higher β value corresponds to a higher ρ_s value. Also notice that changing γ alters the value of ρ_s but not the shape of the response curve. The response curves are linear because we only include the first two terms in our Taylor series expansion.

depositing particles, with values closer to $\beta = 1$ corresponding to a lower repulsive force. If $\beta = 1$, there is not interaction between particles on the slide and depositing particles. Finally, if $\beta > 1$ (orange curve in Figure 2.9), particles on the slide attract depositing particles. This attraction not only allows a coverage density greater than 100%, a scenario we call over-packing, but also produces a maximum particle coverage density. While the particles used in our experiment do not exhibit $\beta > 1$ behavior, the model remains significant in several other applications, including extensions to voter models and other scenarios investigated in the social sciences.

Of note in Figure 2.10 is the linear behavior. This is the result of our use of only the constant and linear terms in our Taylor series approximation for ρ_s. We will discuss the limits of including only these terms in Chapter 4, where we compare our model and simulation results.

2.3 Simulation Techniques and Results

The use of the mean field approximation to convert our CSAE-TL and CSAE-NN transition rates into numerically solvable partial differential equations for particle coverage density depends upon several assumptions. Perhaps the most important is the supposition that each constituent particle experiences the same overall average electric field (i.e., a mean field). To verify this method, we must compare our models' results to experimental data. As we will explore in Chapter 2.4, physical samples of nanoparticle bilayers produced via ISAM are necessarily time consuming to produce. Thus, one efficient way to test the validity of our use of the mean field approximation is via computer simulation.

As originally published in [46], we produce two Python simulations of the ISAM process, one that performs site transitions according the CSAE-TL transition rate (Equation (2.1)) and one that performs site transitions according the the CSAE-NN transition rate (Equation (2.9)). The syntax governing each Python program is shown in detail in Appendix C.3. Here, we discuss the general methodology surrounding our simulation technique, which is based on the ubiquitous Monte Carlo method.

Throughout our simulation efforts, we attempt to replicate attachment and detachment on a slide with $(m \times n)$ attachment sites. Each simulation begins by collecting values for the appropriate evaporation and deposition coefficients ($\gamma, \mu, \alpha, \beta, \eta$, etc.). The program then creates an $(m + 2 \times n + 2)$ matrix:

$$\mathbf{D} = \begin{bmatrix} d_{0,0} & d_{0,1} & \cdots & d_{0,n} & d_{0,n+1} \\ d_{1,0} & d_{1,1} & \cdots & d_{1,n} & d_{1,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{m,0} & d_{m,1} & \cdots & d_{m,n} & d_{m,n+1} \\ d_{m+1,0} & d_{m+1,1} & \cdots & d_{m+1,n} & d_{m+1,n+1} \end{bmatrix}. \tag{2.25}$$

The entries found in submatrix

$$\mathbf{D_{sub}} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,n} \\ \vdots & \ddots & \vdots \\ d_{m,1} & \cdots & d_{m,n} \end{bmatrix} \tag{2.26}$$

represent the sites (n_i) found on the simulated slide. Meanwhile, the exterior entries $\{d_{0,0}, d_{0,1}, \cdots, d_{0,n}, d_{0,n+1}\}$, $\{d_{m+1,0}, d_{m+1,1}, \cdots, d_{m+1,n}, d_{m+1,n+1}\}$, $\{d_{0,0}, d_{1,0}, \cdots, d_{m,0}, d_{m+1,0}\}$, and $\{d_{0,n+1}, d_{1,n+1}, \cdots, d_{m,n+1}, d_{m+1,n+1}\}$ represent nanoparticles within the SiO_2 suspension that are in the same plane as the surface of the slide.[4]

Before simulation begins, every entry in $\mathbf{D_{sub}}$ is assigned a value 0, while every exterior entry is assigned a value of 1. This attempts to replicate the initial state of the system: when the slide is exposed to the nanoparticle suspension, it is completely unoccupied (every entry $d_{i,j} = 0$), while, to a first approximation, the suspension is concentrated enough to be completely occupied (every site in the suspension holds a value of 1). Before every simulated time step, the program first sets every exterior entry to a value of 1.[5] It then cycles through every entry in $\mathbf{D_{sub}}$. At each entry, it first generates a random number $R \in [0, 1]$. Next, it determines whether the entry is occupied ($d_{i,j} = 1$) or unoccupied ($d_{i,j} = 0$). Then, using the appropriate evaporation and deposition coefficients and either the CSAE-TL or CSAE-NN transition rate, the simulation calculates a value $P \in [0, 1]$. When $P \leq R$, the site changes state, representing either evaporation or deposition depending on its original state. When $P > R$, the site maintains its original state. At the end of each time step, the simulation program calculates the simulated slide's coverage density by dividing the number of occupied entries in $\mathbf{D_{sub}}$ by the total number of entries in $\mathbf{D_{sub}}$.

[4] While tracking nanoparticles from the suspension which occupy the same plane as the slide is not particularly important when simulating ISAM, it becomes especially important in our simulations of nanoparticle assembly under parallel electric fields (Chapter 3.3). For this reason, we include these elements in our simulation.

[5] Again, this becomes important under parallel electric fields, when the exterior entries can change state. We assume that the suspension is concentrated enough to ensure that there is always a nanoparticle available on the exterior of the slide.

Our simulations produce several different outputs. The first is a plot of the coverage density versus time step, an example of which is shown in Figure 2.11. Notice how the shape of this simulation output is similar to the shape of the numerical solutions we produced using our mean field approximation-derived rate equations. The slide's particle coverage density rapidly increases until steady state is achieved. However, the steady state in the simulation result shows a slight variance around an average steady state value. This presumably indicates that the simulation is better at demonstrating small fluctuations in particle coverage density as evaporation and deposition continue to occur on the slide's surface after steady state has been achieved.

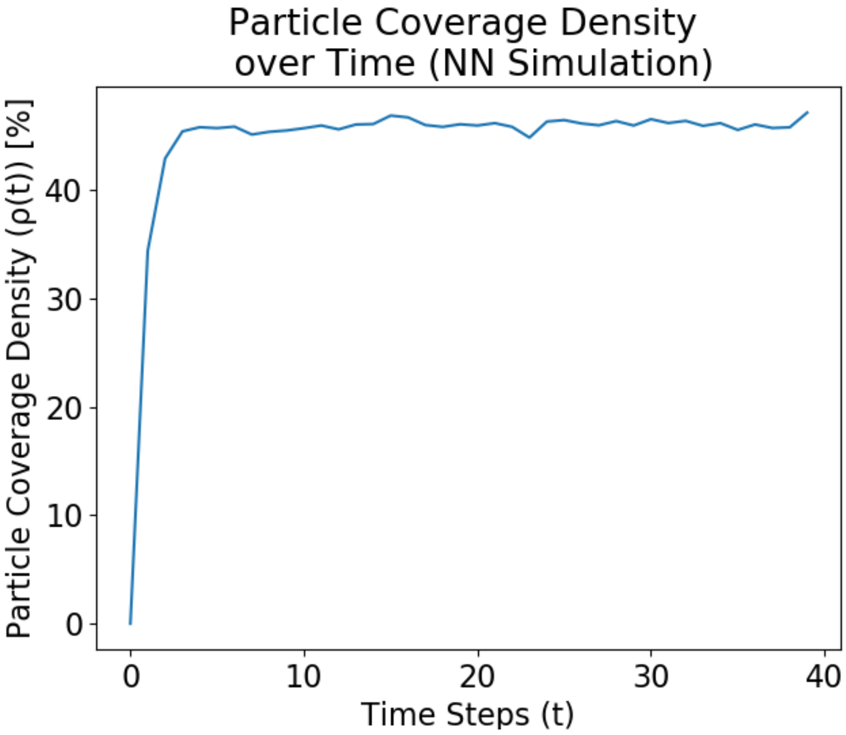

Figure 2.11: Particle coverage density for a (100×100) grid produced using our simulation method with the CSAE-NN transition rate. Notice the rapid approach of steady state, just as predicted by our CSAE-NN model. (Reproduced from [46].)

The second output type is a static image of the slide's state at the end of any time step. These images are produced using MatPlotLib's `imshow` function, which converts entries in a matrix into an image. Figure 2.12 is an example of one of these images. The blue cells represent sites that are occupied by a nanoparticle ($d_{i,j} = 1$), while the white cells represent unoccupied sites ($d_{i,j} = 0$). Note that, like with the matrix $\mathbf{D}$, only the interior cells represent the slide's surface. The exterior cells (all blue/occupied in our example image) represent particles in the suspension.

The third output type is an animation showing how the slide's occupation state changes over time. We produce this animation using the `ArtistAnimation` function from MatPlotLib's `animation` class. While this method is described in more detail in Appendix C.5, we note here that the function simply collects individual static images and plays them in succession. Thus, by producing a static image of the simulated slide's state at the end of each time step, `ArtistAnimation` allows us to easily produce an animation of the entire adsorption process. An example of one of

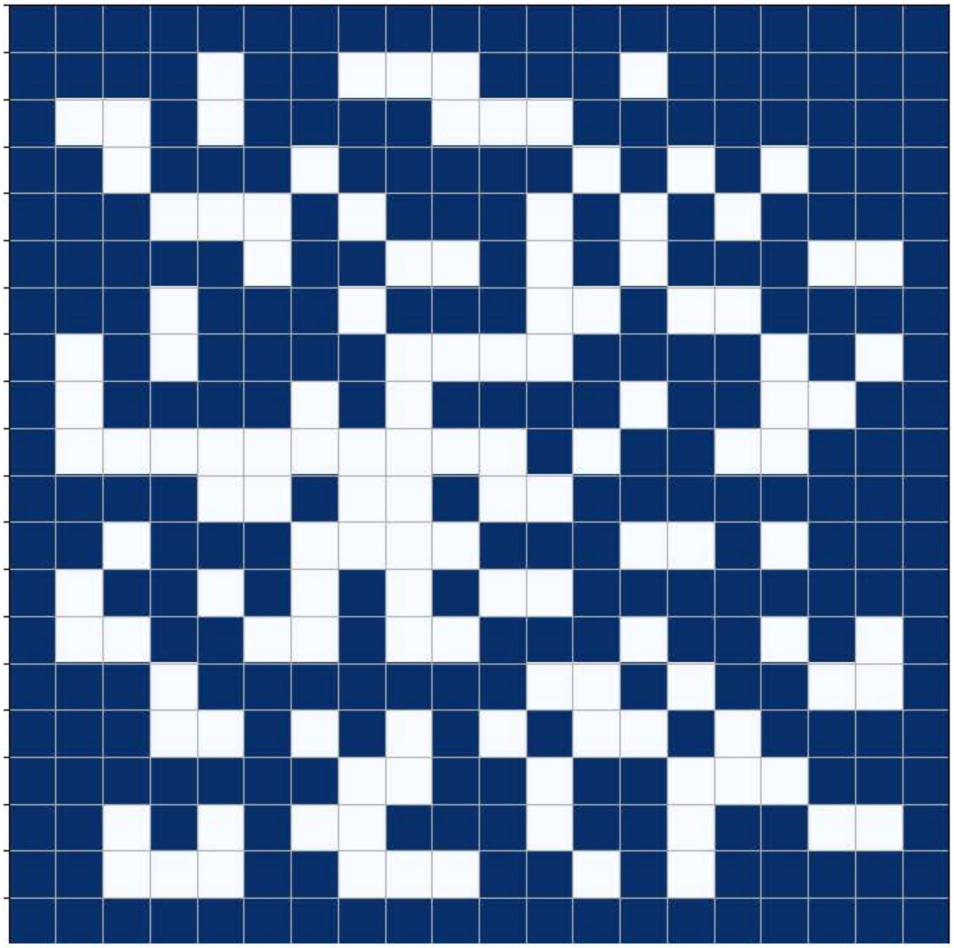

Figure 2.12: A sample (20×20) grid produced via a simulation of the CSAE-NN transition rate. Notice that all sites on the outer edges (the suspension) are occupied (blue), while internal sites (the slide surface) are either occupied or unoccupied (white). (Reproduced from [46].)

our animations can be found at `https://youtu.be/nQiVCYc1epk`.

Figures 2.13 and 2.14 demonstrate how the steady state particle coverage density ρ_s changes as γ and μ vary for our simulation of the CSAE-TL model, while Figures 2.15 and 2.16 demonstrate how ρ_s changes as γ and β vary for our simulation of the CSAE-NN model. We carried out all simulations on a 100×100 grid over 1000 time steps. In Chapter 4, we compare in detail the results of our models (Figures 2.5 - 2.6 and 2.9 - 2.10) and our simulations.

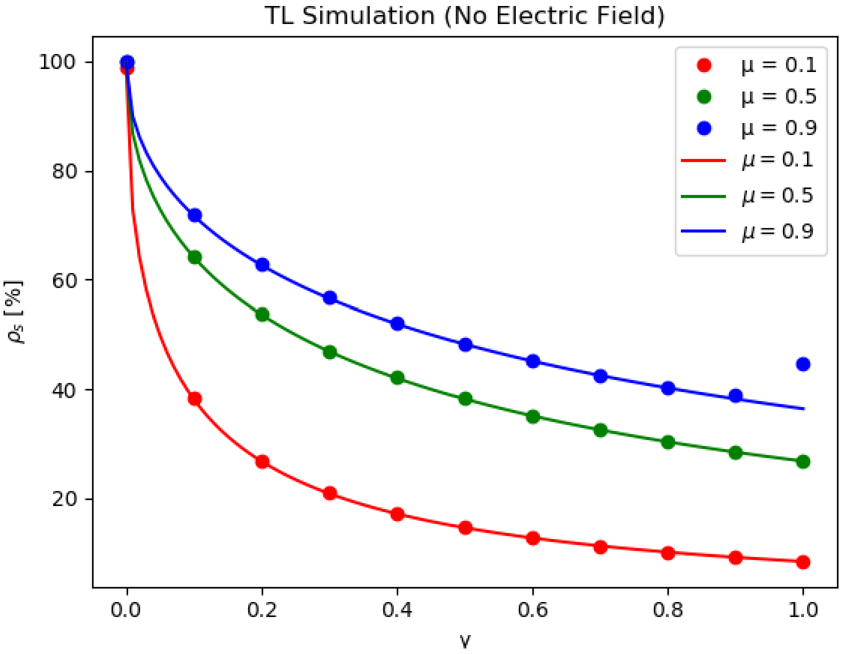

Figure 2.13: The simulated γ response (points) overlaid with the model-predicted γ response (curves) for CSAE-TL.

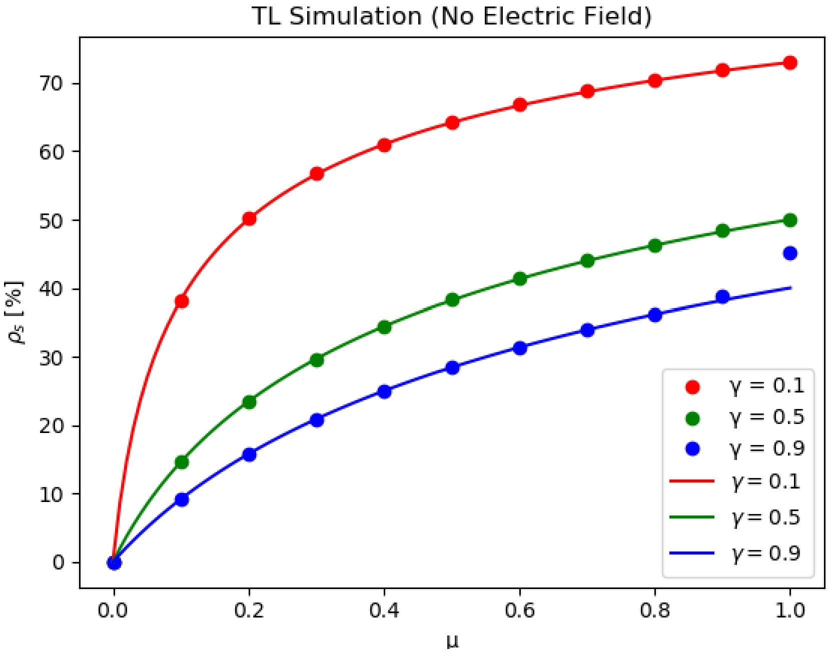

Figure 2.14: The simulated μ response (points) overlaid with the model-predicted μ response (curves) for CSAE-TL.

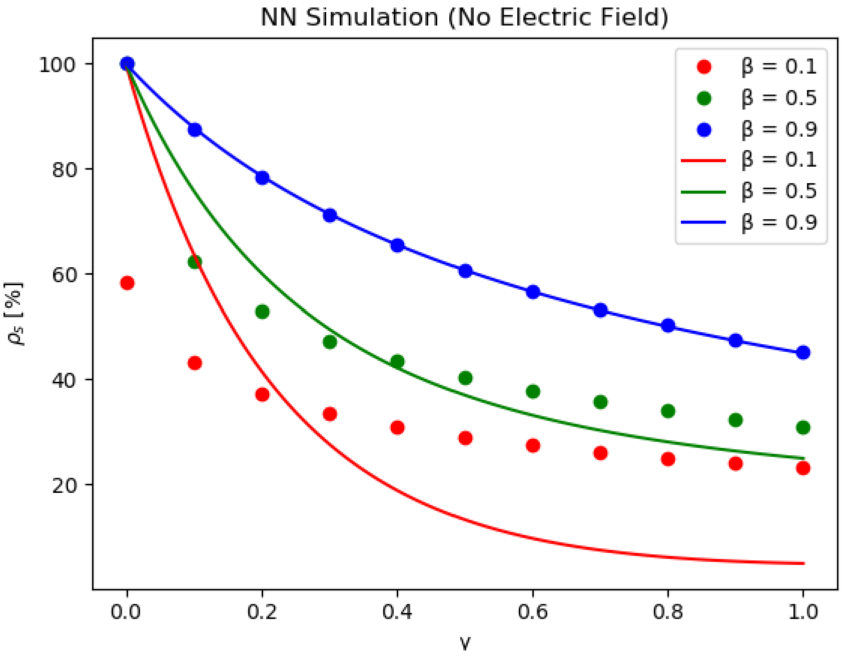

Figure 2.15: The simulated γ response (points) overlaid with the model-predicted γ response (curves) for CSAE-NN.

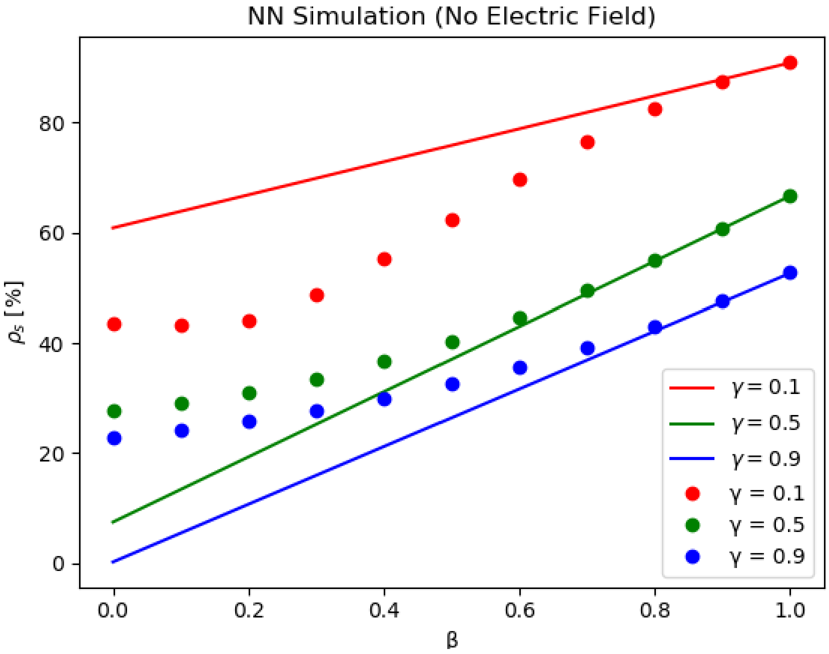

Figure 2.16: The simulated β response (points) overlaid with the model-predicted β response (curves) for CSAE-NN.

2.4 Experimental Techniques and Results

Using ISAM to produce thin films in the laboratory setting is a time consuming process. Before sample production, we first prepare suspensions of PDDA and SiO_2 nanoparticles by mixing stock suspension with de-ionized (DI) water. We determine the ideal concentration for each investigation via a series of qualitative experiments, in which we image samples produced using a wide variety of PDDA and SiO_2 concentrations. We select the combination of concentrations which result in the most uniform distribution of particles. We allow every suspension to stir for at least 24 hours before deposition. This stirring helps to ensure a uniform distribution of the stock (solute-like material) throughout the DI water (solvent-like material).

After preparing our suspensions, we clean our glass microscope slides using the Acetone-Methanol-Isopropol Alcohol (AMI) method [49]. We place each slide in a staining jar filled with acetone and allow it to soak under sonication for five minutes. We then move the slides to a staining jar filled with methanol for another five minutes of sonication, followed by a staining jar of isopropol alcohol for yet another sonication cycle. The now-cleaned slides finally move to a DI water bath, where they remain until they are dried under a gentle flow of N_2 gas. Drying under a flow of N_2 gas prevents spotting, as well as the introduction of dust particles from the atmosphere.

Sample production then proceeds as follows: for each sample, we fill one staining jar with unused PDDA suspension, one with unused SiO_2 suspension, and two with unused DI water. We dip a cleaned slide in the PDDA suspension, allowing it to soak. We then quickly move the slide to one of the two DI water jars, where we gently stir it. Following this rinsing step, we dip the slide in the SiO_2 jar, where it again soaks. We rinse the slide in the remaining DI water jar, leaving it in the DI water until we are ready to dry the sample under a gentle flow of N_2 gas. As during the cleaning procedure, the N_2 gas prevents spotting as well as the deposition of particulates from the atmosphere. Figure 2.1 shows this process schematically. The time periods used for each dipping and rinsing step vary depending upon the requirements of each investigation.

Determining the steady state particle coverage density of each sample requires the production of several images of the sample's surface. Throughout our investigation, we use scanning electron microscopy (SEM) as our imaging method of choice. SEM requires conductive samples. Therefore, before we image our samples, we first coat them in gold using a Cressington 108 plasma sputter coater. Under vacuum, the sputter coater bombards a thin sheet of gold with argon atoms, producing a plasma which uniformly coats the sample.

We coat each of our samples two times for one minute. We find that this coating procedure produces clear, analyzable SEM images of our particles. We use carbon tape to mount the sample onto the SEM stage, ensuring that a portion of the tape wraps around to the surface we intend to image. Wrapping the tape onto the imaging surface provides a conductive path for electrons that often build up on the surface of the slide. This buildup of electrons, often called "charging" of the sample, reflects electrons released from the SEM's electron gun, obscuring the image.

We image each gold coated sample using a EVO MA-15 scanning electron microscope, manufactured by Zeiss. Our images are produced via the detection of secondary electrons in a high vacuum. In particular, we use the specifications listed in Table 2.1. To ensure that we capture a representative sample of particle coverage densities across the entire slide, we often perform a stratified random sample, in which we divide the slide into 16 individual strata and use a random number generator to determine which strata to image.

SEM Setting	**Value**
EHT Target	20.00 kV
Aperture Size	20.00 μm
Collector Bias	300 V
Signal	SE1
Spot Size	150

Table 2.1: SEM settings used throughout our experimental investigation.

Figure 2.17 is one of the many SEM images we produced throughout our investigation. Notice how the nanoparticles in this image exhibit high contrast with the background, a characteristic that will aid in our calculation of the particle coverage density. To determine particle coverage density, we use ImageJ, a Java-based image processing and analysis suite originally designed by the National Institutes of Health. Due to the high contrast in our SEM images, we can easily use ImageJ's `Make Binary` process to convert our image into a black and white equivalent without the loss of significant detail (see Figure 2.18).[6] The program's `Analyze Particles` process then easily identifies the particles and calculates the particle coverage density. Figure 2.19 is an example of the visual output provided by the `Analyze Particles` process.

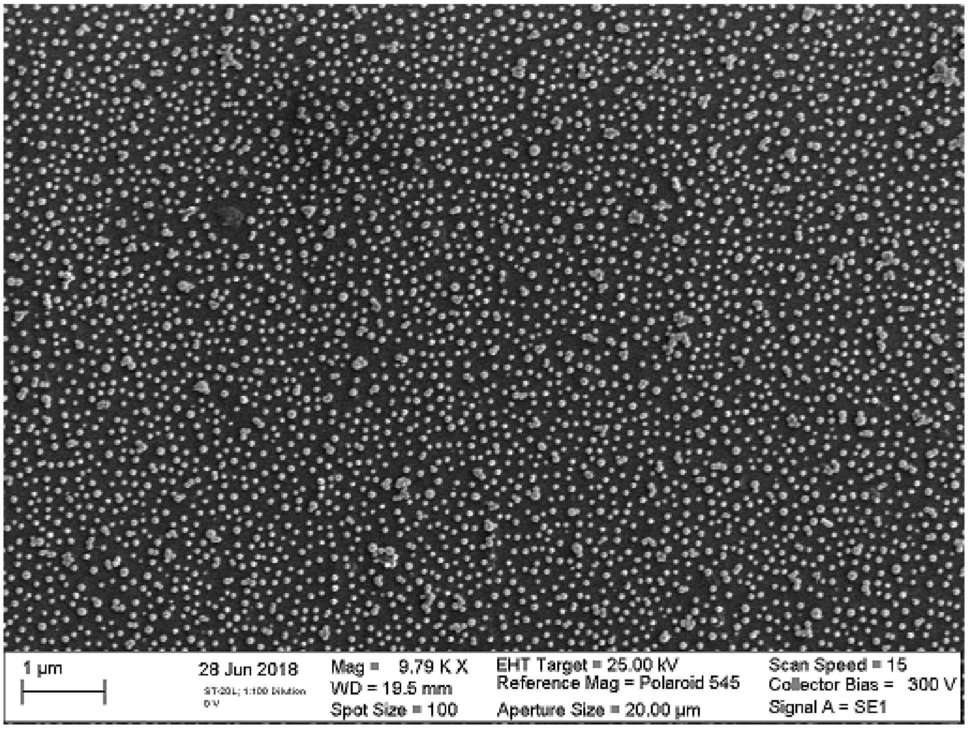

Figure 2.17: An unprocessed SEM image of a 1:100 dilution of ST-20L SiO_2 nanoparticles. The image is magnified 9.79 thousand times.

[6] We also `Invert` the image to ensure that the particles are represented by black regions while the slide surface is represented by white regions.

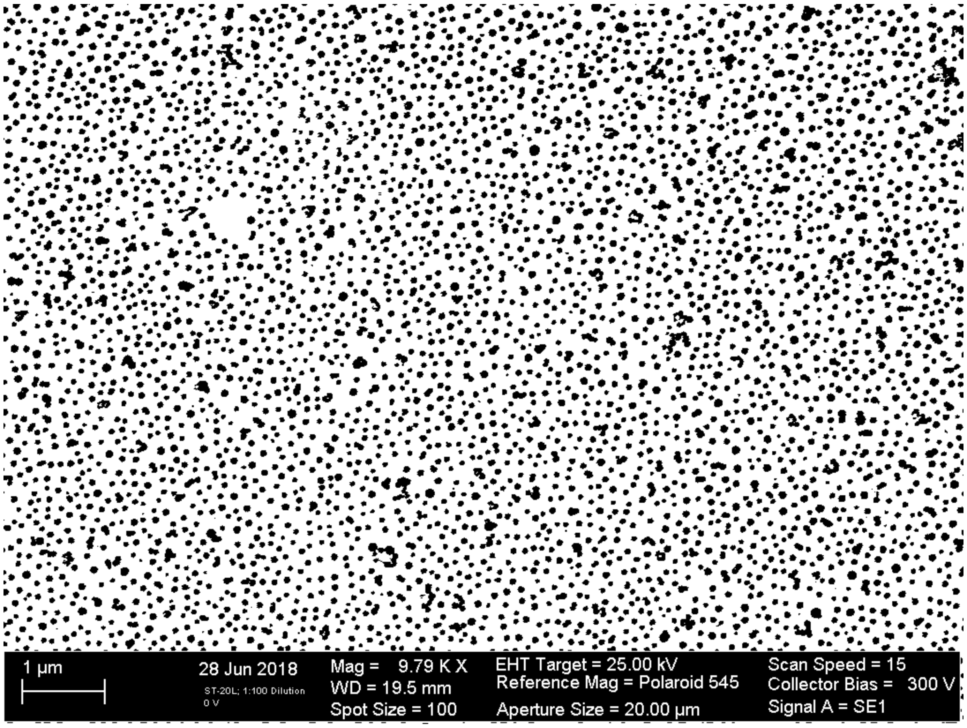

Figure 2.18: The same SEM image after using ImageJ's `Make Binary` and `Invert` tools.

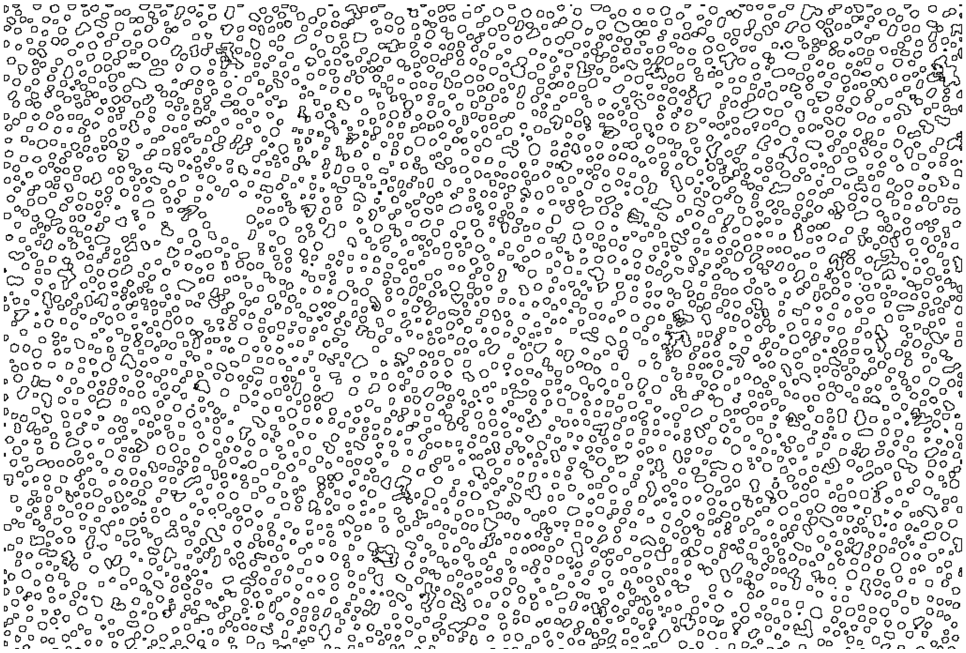

Figure 2.19: Particle outlines detected using ImageJ's `Analyze Particles` tool. The tool also calculates the image's particle coverage density.

The results of our experiments are published in [50]. Our first investigation focuses on determining the relationship between a slide's steady state particle coverage density and the concentration of the nanoparticle suspension. We produce and image samples for each suspension concentration using the methods outlined above. Table 2.2 provides an overview of the experimental conditions, and Figure 2.20 summarizes our results.

Parameter	**Value**
SiO_2 Nanoparticles	SNOWTEX ST-20L from Nissan Chemical
Suspension pH	10.3
Suspension Temperature	21 $^{\circ}$C
PDDA Concentration	10 mM
Dipping Time (per step)	10 min
Rinsing Time (per step)	1 min

Table 2.2: Experimental conditions during our investigation of the relationship between particle coverage density and suspension concentration.

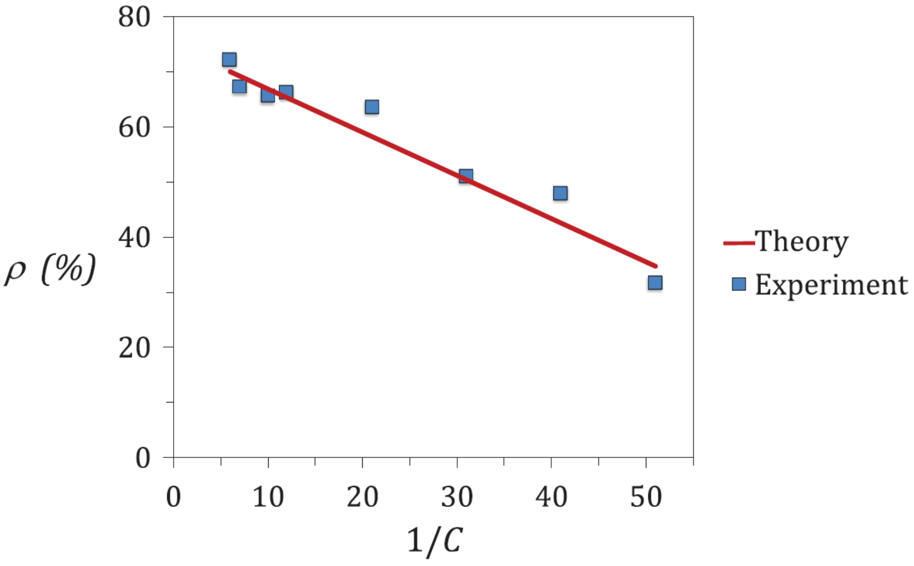

Figure 2.20: Experimental and theoretical concentration dependence of particle coverage density. Notice the strong agreement between our experimental data and theoretical predictions. Also notice that ρ and $1/C$ are linearly related. (Reproduced from [50].)

By fitting the CSAE-TL model to our experimental results (see red curve in Figure 2.20), we find that we can express the deposition coefficient μ in terms of the evaporation coefficient γ and concentration C as follows:

$$\mu = e^{-\gamma/2C}. \tag{2.27}$$

This equation maintains its validity when γ is low ($\gamma \approx 0.1$). It generally performs better at low concentrations.

Our second investigation seeks a thorough description of the time dependence of the deposition process. While previous investigations [27, 36] thoroughly describe the kinematics of nanoparticle deposition, they do not provide an adequate investigation of the time scales necessary to achieve steady state. Theoretically, electrostatic screening causes the deposition process to divide into two distinct regimes: a rapid Langmuir-type adsorption followed by a slow approach to the maximum particle coverage density. [27] suggests a 90% saturation of a PDDA monolayer with adsorbed silica in 10 s. However, because these results were collected over longer time periods (on the order of seconds), they failed to capture information about how full saturation is achieved. By imaging particle coverage densities that occur during deposition times less than 2 s, we hope to characterize this portion of the deposition process, especially the amount of time needed to transition from Langmuir-type adsorption to slow adsorption.

To obtain this characterization, we begin by dipping cleaned slides in a 10 mM suspension of PDDA for 10 min. We follow the PDDA dip with a 1 min rinse in DI water and then dry the slides under a stream of N_2 gas. Each slide is pre-marked with small, regularly spaced reference dots. As we dip them into the SiO_2 suspension, we film the process with a high-speed camera at 1000 frames/s. This allows us to review the footage and determine the exact time each pre-marked point spends in the suspension, to a precision on the order of milliseconds. We collect 11 data points, ranging in exposure time from 0.058 s to 0.639 seconds. We image each using the SEM and calculate the coverage density using ImageJ.

Figure 2.21 shows our experimental results, along with the predictions of the CSAE-TL model ($\mu = 0.7$ and $\gamma = 0.2$). The results show that approximately half of the particles are deposited in the first 0.058 s. This corresponds with the rapid Langmuir-type adsorption. Between 0.058 s and 0.639 s, adsorption continues slowly, and the system eventually transitions smoothly to steady

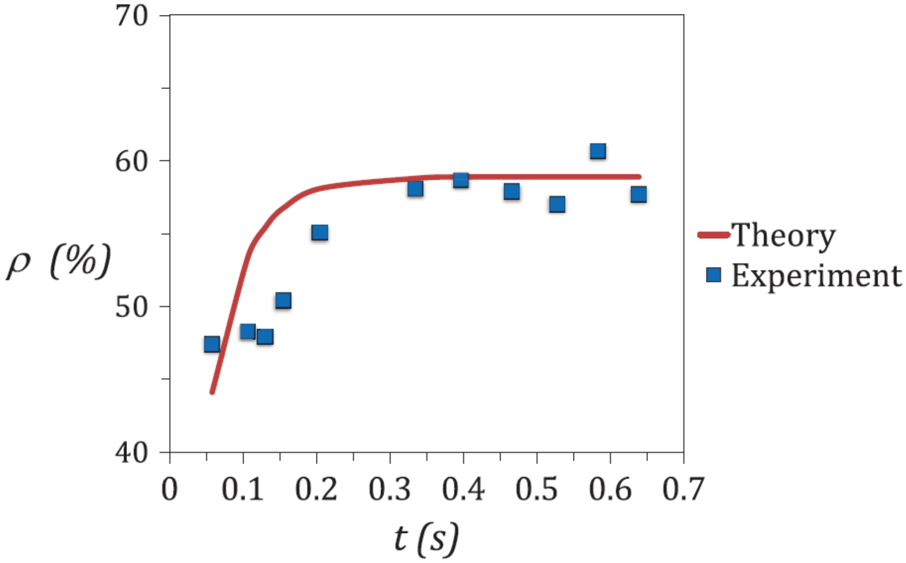

Figure 2.21: Experimental and theoretical time dependence of particle coverage density. Notice that half the particles are deposited in the first 0.058 s (Langmuir-type adsorption). The remainder deposit during a smooth transition to steady state. Also notice the strong agreement between our theoretical predictions and experimental data. (Reproduced from [50].)

state. This time interval corresponds to the expected slow approach to maximum particle coverage density. Overall, the system achieves steady state much faster than in [27]. We believe this occurs because their results were produced using a nanoparticle suspension containing NaCl, while our suspension lacked NaCl. When NaCl is present, the attractive force between the PDDA and SiO_2 molecules is reduced since counterions can form. This reduced attractive force should slow down the deposition process.

Chapter 3

Self-Assembly under External Electric Fields (DSAM)

While the ability to model, simulate, and produce thin film samples via ISAM is of great importance to nanoscientists, there remains much to be learned about the effect of external bias on the assembly process. Particles could undergo self-assembly in a variety of environments, including those with strong electric fields, strong magnetic fields, or even intense mechanical vibrations. In this chapter, we explore attempts to model, simulate, and produce thin films formed under the influence of electric fields. In doing so, we develop a new production technique, dubbed Directed Self-Assembly of Monolayers, or DSAM, in which the particle coverage density of a thin film sample is controlled by the strength of an applied electric field. The ability to easily control particle coverage density promises to be of great importance to researchers and manufacturers alike.

3.1 Overview of Electric Fields, Electric Potentials, and Capacitors

Before discussing two distinct electric-field driven DSAM techniques, we first briefly review the physics of electric fields and potentials [51], as well as parallel plate capacitors [52]. Like any classical field, an electric field pervades all space. It originates from any object that possesses electric charge, typically denoted with the variable Q. Likewise, only an object with an electric charge may experience an electric field. Mathematically, the electric field is perhaps best understood in relation to the electric force $\boldsymbol{F_e}$. In general, we understand that a particle with charge Q experiences a force

$$\boldsymbol{F_e} = Q\boldsymbol{E} \tag{3.1}$$

when subjected to an electric field $\boldsymbol{E}$. The electric field itself may be calculated one of two ways. If originating from a collection of point-like charges, the principle of superposition states that the field may be calculated by adding the field vectors of each individual charge. Mathematically, this technique is written as

$$\boldsymbol{E}(\boldsymbol{r}) \equiv \frac{1}{4\pi\varepsilon_0}\sum_{i=1}^{n}\frac{q_i}{\imath_i^2}\hat{\boldsymbol{\imath}}_{\boldsymbol{i}}, \tag{3.2}$$

where ε_0 is the permittivity of free space, q_i is the charge of point-like charge i, and $\boldsymbol{\mathscr{r}_i} = \boldsymbol{r} - \boldsymbol{r}_i'$, as depicted in Figure 3.1. If originating from a continuous distribution of charge, the field may be calculated via integration:

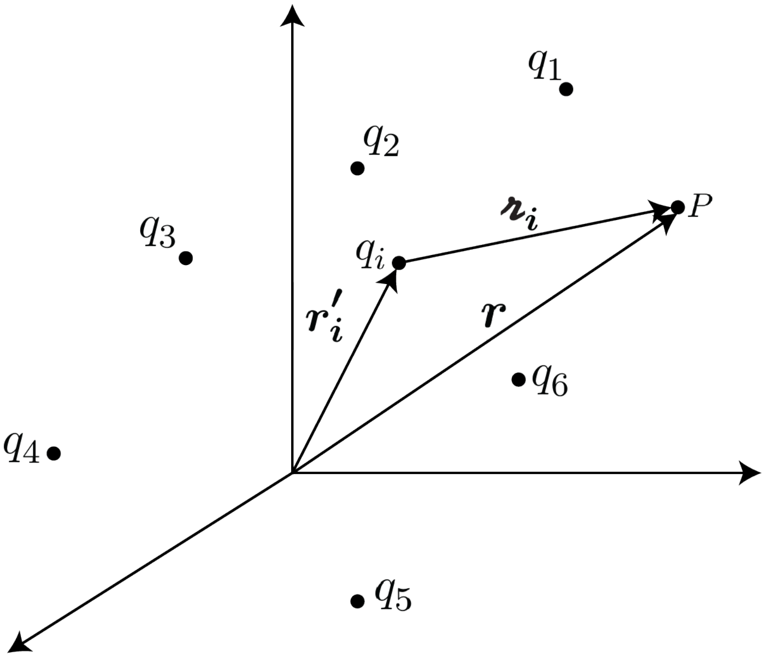

Figure 3.1: Calculating the electric field at point P produced by a collection of charges q_i. Calculating the position vector of P in the frame of each charge q_i allows us to sum the individual electric fields. This sum represents an effective field felt by any charged object residing at point P.

$$\boldsymbol{E}(\boldsymbol{r}) = \frac{1}{4\pi\varepsilon_0}\int \frac{1}{\mathscr{r}^2}\hat{\boldsymbol{\mathscr{r}}}dq. \tag{3.3}$$

dq can be written as

$$\begin{aligned} dq &\to \lambda dl' \quad (3.4)\\ &\to \sigma da' \quad (3.5)\\ &\to \rho d\tau', \quad (3.6)\end{aligned}$$

where λ is the charge-per-unit-length, σ is the charge-per-unit-area, and ρ is the charge-per-unit-volume, for linear (dl'), surface (da'), and volume ($d\tau'$) charge distributions, respectively.

Since the curl of an electrostatic field $\boldsymbol{E}$ is always 0 (i.e., $\nabla \times \boldsymbol{E} = 0$), the electrostatic field is conservative and possesses a potential function $V(\boldsymbol{r})$. We define the electric potential as

$$V(\boldsymbol{r}) \equiv -\int_O^{\boldsymbol{r}} \boldsymbol{E}\cdot d\boldsymbol{l}, \tag{3.7}$$

where O is some predefined reference point. Using this definition of electric potential, we say that there exists a potential difference between two points $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$V(\boldsymbol{b}) - V(\boldsymbol{a}) = -\int_{\boldsymbol{a}}^{\boldsymbol{b}} \boldsymbol{E}\cdot d\boldsymbol{l}. \tag{3.8}$$

The potential difference plays an important role in the calculation of the field of a parallel plate capacitor, the primary tool we use to produce electric fields during DSAM. Additionally, we can easily calculate the electric field produced by an electric potential via

$$\boldsymbol{E} = -\nabla V. \tag{3.9}$$

Equation (3.9) suggests that electric field lines are always perpendicular to the contours produced by the potential.

One common method of producing a constant electric field $\boldsymbol{E}$ is the use of a parallel plate capacitor. As depicted in Figure 3.2, a parallel plate capacitor produces a uniform, linear electric field, directed from the plate with higher electric potential to the plate with lower electric potential.[1] The magnitude of an electric field $\boldsymbol{E}$ produced by a parallel plate capacitor is given by

$$E = \frac{V}{d}, \tag{3.10}$$

where V is the potential difference between the plates and d is the distance between the plates.

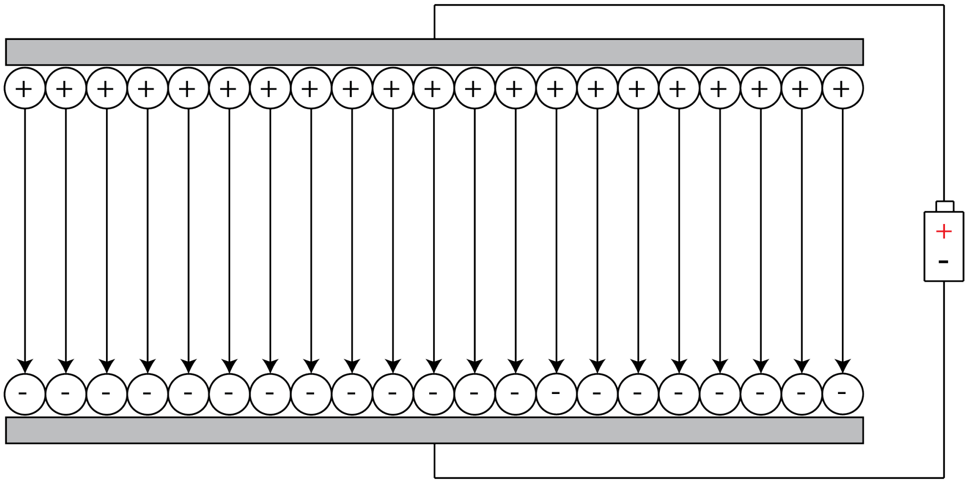

Figure 3.2: The field lines within a parallel plate capacitor attached to a DC source. The electric field always flows from the positively charged plate to the negatively charged plate.

3.2 Perpendicular Electric Fields

When applying an electric field during the ISAM process, the field's orientation is critical to determining the modeling, simulation, and experimental techniques used. While any orientation is theoretically possible, we limit our discussion to fields that are perpendicular and parallel to the surface of the glass slide used during the deposition process. In this section, we describe our efforts to model, simulate, and produce experimentally samples that are subjected to perpendicular electric fields. The modeling portion of our treatment is published as [53]. This scenario is by far the simplest, as negatively charged particles will be forced directly onto the slide when subjected

[1] Near the edges of the plates, the field is not truly uniform. However, since we only consider effects near the center of the capacitor throughout our work, we will treat the field as if it were perfectly uniform.

to a field oriented out of the surface of the slide. Conversely, they will be forced away from the slide when subjected to a field oriented toward the slide's surface.

3.2.1 DSAM for Perpendicular Fields

The process for achieving directed self-assembly under a perpendicular electric field is similar to the traditional ISAM process (see Figure 2.1). We again begin by dipping a clean microscope slide, which possesses and innate negative surface charge, in the standard polycation PDDA. This gives the slide a positive surface charge, which will allow negatively charged SiO_2 nanoparticles to adhere to its surface. Just before dipping in the SiO_2 suspension, however, we power a parallel plate capacitor, which is placed on the outside of the staining jar holding the suspension. As depicted in Figure 3.3, the plates of the capacitor are oriented parallel to the surface of the glass slide. This orientation produces an electric field that is perpendicular to the slide's surface. For a field that is directed away from the slide's surface, which encourages negatively charged particles to travel toward the glass slide, the negative terminal of the power supply is attached to the plate closest to the surface to be examined. Conversely, for a field that is directed toward the slide's surface, which

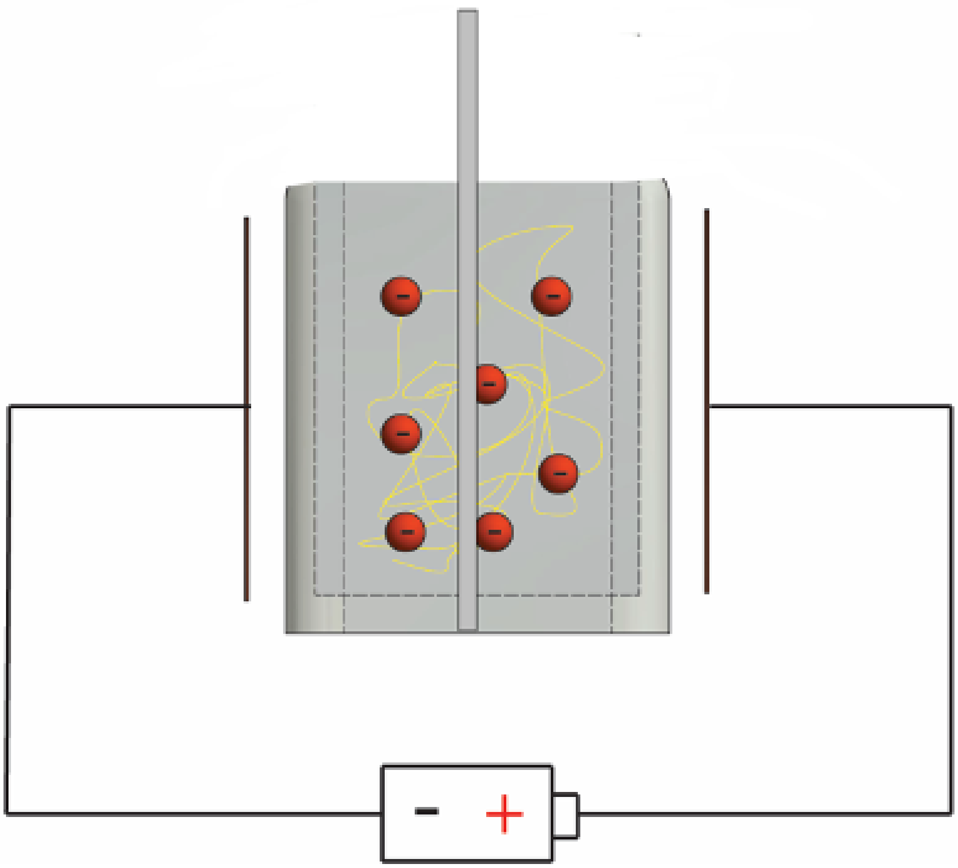

Figure 3.3: Our modification of the nanoparticle staining jar to allow for the application of a uniform perpendicular electric field across the slide face during self-assembly. Note that the two capacitor plates (indicated by the two vertical lines on either side of the jar) are actually adhered to the side of the jar. The separation in the image is for visual clarity only.

encourages negatively charged particles to travel away from the glass slide, the positive terminal of the power supply is attached to the plate closest to the surface to be examined. We leave the capacitor powered until the end of the deposition process, ensuring that depositing/evaporating nanoparticles experience the effects of the uniform field at all times.

3.2.2 CSAE Modeling Techniques and Results

Like with ISAM, we once again use cooperative sequential adsorption with evaporation (CSAE) models to represent the assembly process mathematically. To account for the presence of the electric field, however, we alter our transition rates by replacing the evaporation term $n_i\gamma$ with a new set of terms: $\gamma_1(1-n_i)+\gamma_2(n_i)$. The $\gamma_1(1-n_i)$ term represents particles being driven onto the surface of the slide (with probability γ_1) by the electric field. Similarly, the $\gamma_2(n_i)$ term describes particles being driven away from the surface of the slide (with probability γ_2) by the electric field. Both γ_1 and γ_2 can be scaled to the strength of the applied electric field (typically given in V/m) using experimental data, with higher values of γ_1 and γ_2 corresponding to stronger electric fields.

With our new terms implemented, the transition rate for the CSAE-TL model becomes

$$c_{TL}(n_i \to (1-n_i)) = \gamma_1(1-n_i)+\gamma_2(n_i)+\mu(1-n_i)\left(1-\frac{\sum_{i=1}^{n} n_i}{N}\right). \tag{3.11}$$

Equation (3.11) can be described via the partial differential equation

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\gamma_2\langle n_i\rangle+\gamma_1(1-\langle n_i\rangle)+\mu\left\langle(1-n_i)\left(1-\frac{\sum_{i=1}^{n} n_i}{N}\right)\right\rangle. \tag{3.12}$$

Applying the mean field approximation, in which higher order correlations are approximated as $\langle n_i n_j\rangle = \langle n_i\rangle\langle n_j\rangle$ and $\rho = \sum_i \frac{n_i}{N}$, yields

$$\frac{\partial\rho}{\partial t} = -\gamma_2\rho+\gamma_1(1-\rho)+\mu(1-\rho)^2, \tag{3.13}$$

the mean field equation for the CSAE-TL model under the influence of perpendicular electric fields. The steady state solution ρ_s to this equation, found when $\frac{\partial\rho}{\partial t}=0$, is then

$$\rho_s = \frac{\pm\sqrt{\gamma_1^2+2\gamma_1\gamma_2+4\gamma_2\mu+\gamma_2^2}+\gamma_1+\gamma_2+2\mu}{2\mu} \tag{3.14}$$

when $\mu \neq 0$ and

$$\rho_s = \frac{\gamma_2}{\gamma_1+\gamma_2} \tag{3.15}$$

when $\mu = 0$ and $\gamma_1+\gamma_2 \neq 0$.

Likewise, the transition rate for the CSAE-NN model becomes

$$c_{NN}(n_i \to (1-n_i)) = \gamma_1(1-n_i)+\gamma_2(n_i)+(1-n_i)\alpha\beta^{\eta}. \tag{3.16}$$

Equation (3.16) can be described via the partial differential equation

$$\frac{\partial \langle n_i \rangle}{\partial t} = -\gamma_2 \langle n_i \rangle + \gamma_1 (1 - \langle n_i \rangle) + \langle (1 - n_i) \alpha \beta^{\eta} \rangle. \tag{3.17}$$

Using the mean field approximation, Equation (3.17) becomes

$$\frac{\partial \rho}{\partial t} = -\gamma_2 \rho + \gamma_1 (1 - \rho) + (1 - \rho) \alpha \beta^{z\rho}. \tag{3.18}$$

Like with ISAM, the steady state equation for CSAE-NN is transcendental. We can again use a Taylor series expansion about $\beta = 1$ to find an approximate solution, yielding

$$\rho_s = \frac{\alpha + \gamma_1}{\alpha + \gamma_1 + \gamma_2} - (1 - \beta) \left[4 \left(\frac{\sqrt{\alpha^2 - \alpha \gamma_1}}{\alpha + \gamma_1 + \gamma_2} \right)^2 \left(1 - \frac{\alpha + \gamma_1}{\alpha + \gamma_1 + \gamma_2} \right) \right]. \tag{3.19}$$

Like with ISAM, Equations (3.13) and (3.18) can be solved numerically using `odeint`. Figures 3.4 and 3.5 show how steady state particle coverage density changes with respect to μ (for the CSAE-TL case) and β (for the CSAE-NN case) at several different values of $\gamma_2 \neq 0$ when $\gamma_1 = 0$. This configuration of γ-values represents a field directed into the slide, which encourages negatively charged nanoparticles to detach. Figures 3.6 and 3.7 show similar information when $\gamma_1 \neq 0$ and $\gamma_2 = 0$. This configuration of γ-values represents a field directed away from the slide, which encourages negatively charged nanoparticles to attach.

For both CSAE-TL and CSAE-NN, we see the steady state coverage density ρ_s increase as the deposition coefficients μ and β increase. Like in our modeling of assembly under no electric field, these results mirror reality, in which a higher tendency to deposit corresponds to a higher steady state coverage density. In Figures 3.4 and 3.5, where $\gamma_1 = 0$ and $\gamma_2 \neq 0$, we see a higher γ_2 value correspond to a lower ρ_s value. This is because this scenario reflects a field oriented so as to encourage detachment. In Figures 3.6 and 3.7, where $\gamma_1 \neq 0$ and $\gamma_2 = 0$, both models predict that a 100% coverage density will be achieved for all values of μ. In this case, a 100% coverage density is achieved because $\gamma_2 = 0$ ensures that no particles can detach. A scenario in which a field encouraging detachment is superimposed over a system that also experiences evaporation would require a non-zero γ_2 in addition to the non-zero γ_1.

The linear behavior in Figure 3.5 is once again the result of our Taylor series approximation for ρ_s, in which we take only the constant and linear terms. Like with the no-field case, we will discuss the limits of this approximation in Chapter 4.

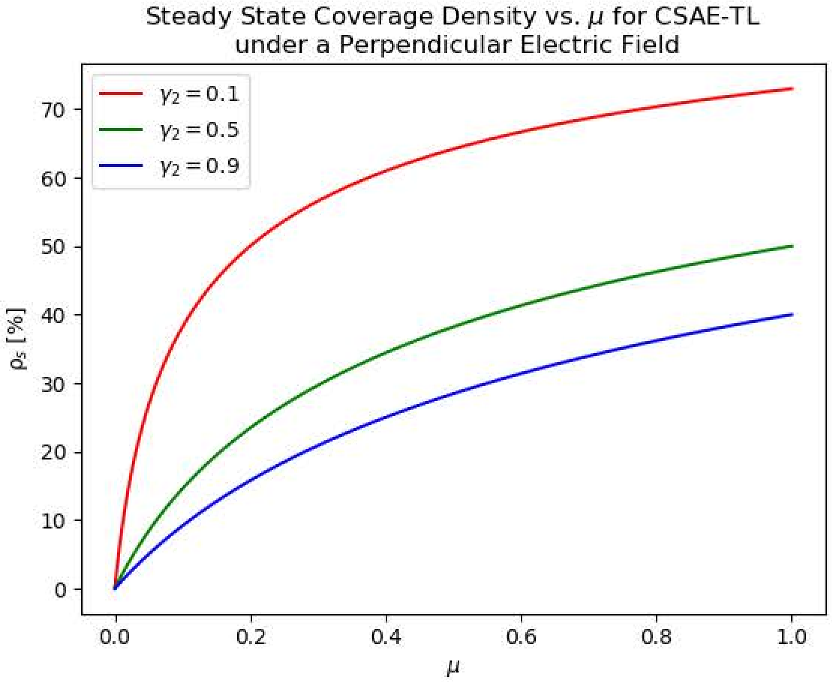

Figure 3.4: The μ response of steady state particle coverage density calculated using our CSAE-TL model. Here, particles assemble under a perpendicular field encouraging particle detachment ($\gamma_1 = 0$, $\gamma_2 \neq 0$). Notice that a higher μ value corresponds to a higher ρ_s value. Also notice that changing γ alters the value of ρ_s but not the shape of the response curve.

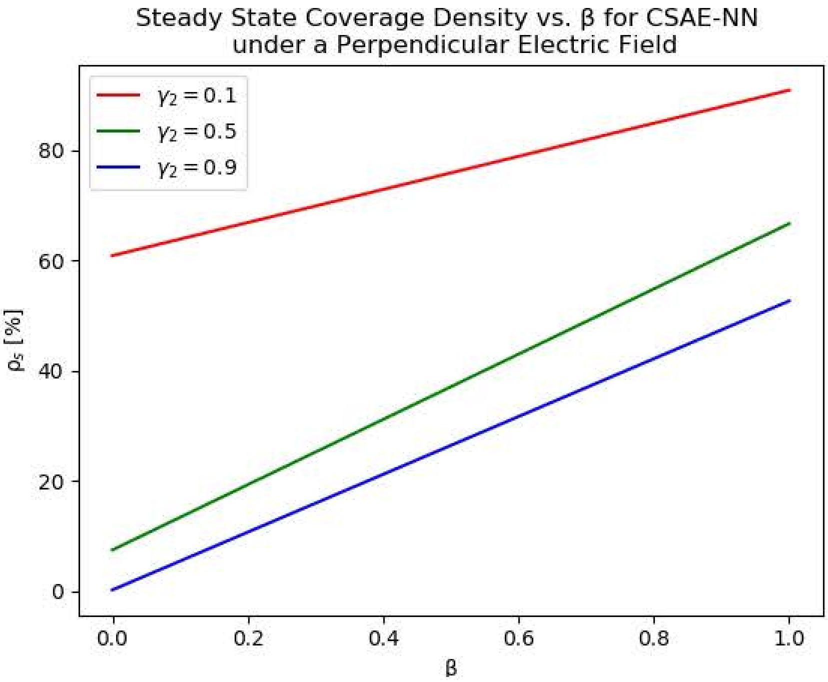

Figure 3.5: The β response of steady state particle coverage density calculated using our CSAE-NN model. Here, particles assemble under a perpendicular field encouraging particle detachment ($\gamma_1 = 0$, $\gamma_2 \neq 0$). Notice that a higher β value corresponds to a higher ρ_s value. Also notice that changing γ alters the value of ρ_s but not the shape of the response curve. The response curves are linear because we only include the first two terms in our Taylor series expansion.

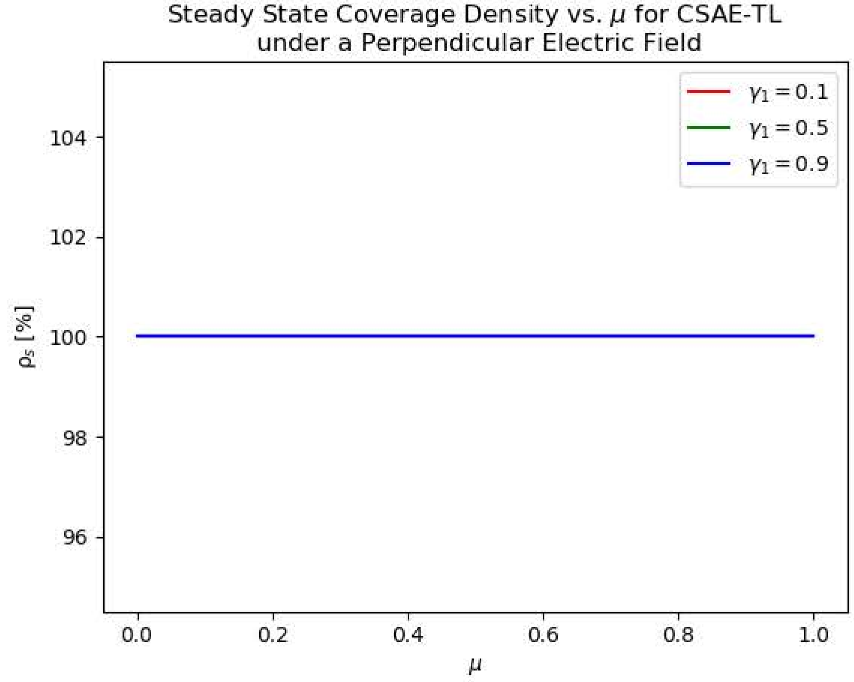

Figure 3.6: The μ response of steady state particle coverage density calculated using our CSAE-TL model. Here, particles assemble under a perpendicular field encouraging particle attachment ($\gamma_1 \neq 0$, $\gamma_2 = 0$). Notice that the system achieves a steady state particle coverage density of $\rho_s = 100\%$ for all γ_1 values since a field of any strength will continue to drive particles onto the slide surface until 100% coverage is achieved. The red and green curves for $\gamma_1 = 0.1$ and $\gamma_1 = 0.5$ are hidden under the blue curve representing $\gamma_1 = 0.9$.

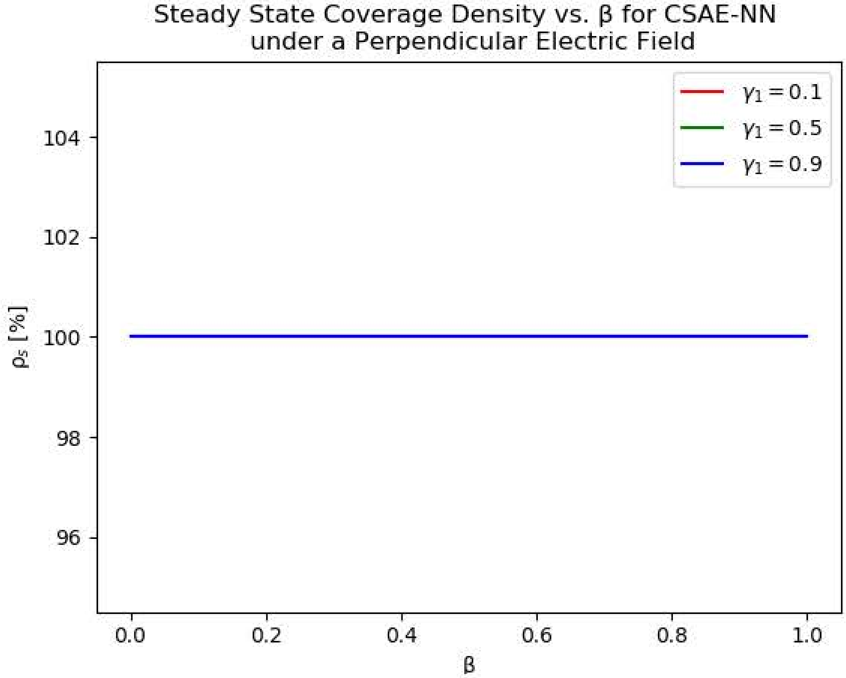

Figure 3.7: The β response of steady state particle coverage density calculated using our CSAE-NN model. Here, particles assemble under a perpendicular field encouraging particle attachment ($\gamma_1 \neq 0$, $\gamma_2 = 0$). Notice that the system achieves a steady state particle coverage density of $\rho_s = 100\%$ for all γ_1 values since a field of any strength will continue to drive particles onto the slide surface until 100% coverage is achieved.. The red and green curves for $\gamma_1 = 0.1$ and $\gamma_1 = 0.5$ are hidden under the blue curve representing $\gamma_1 = 0.9$.

3.2.3 Simulation Techniques and Results

The simulation technique for particles assembling under a perpendicular electric field is identical to the Monte Carlo procedure used for particles assemblying under no field. The only change in the simulation program is the use of the perpendicular electric field CSAE-TL and CSAE-NN transition rates for calculation of the value P, which we use to determine whether a particle site should transition states via comparison to a random number R. Figures 3.8 and 3.9 show steady state coverage densities across a wide range of μ and β values, respectively, from simulations on a 100 × 100 grid over 1000 time steps when $\gamma_1 = 0$ and $\gamma_2 \neq 0$. These values represent a field oriented so as to encourage particle detachment. Figures 3.10 and 3.11 show steady state coverage densities across of a wide range of μ and β values, respectively from simulations on a 100 × 100 grid over 1000 time steps when $\gamma_1 \neq 0$ and $\gamma_2 = 0$. These values represent a field oriented so as to encourage particle deposition. In Chapter 4 we compare these simulation results to our CSAE models for nanoparticle self-assembly under perpendicular electric fields (Figures 3.4 - 3.7).

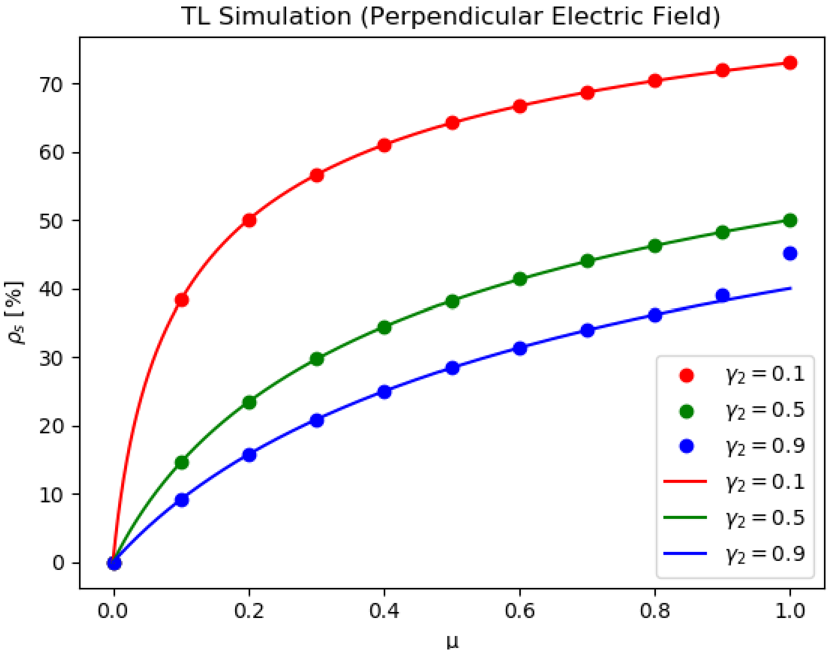

Figure 3.8: The simulated μ response (points) overlaid with the model-predicted μ response (curves) for CSAE-TL. Here, particles assemble under a perpendicular field encouraging particle detachment ($\gamma_1 = 0$, $\gamma_2 \neq 0$).

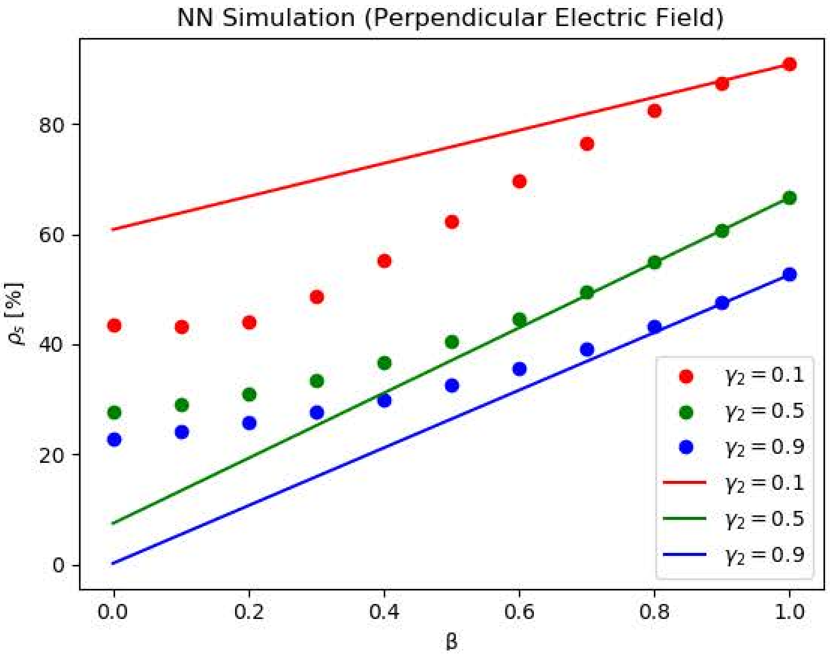

Figure 3.9: The simulated β response (points) overlaid with the model-predicted β response (curves) for CSAE-NN. Here, particles assemble under a perpendicular field encouraging particle detachment ($\gamma_1 = 0$, $\gamma_2 \neq 0$).

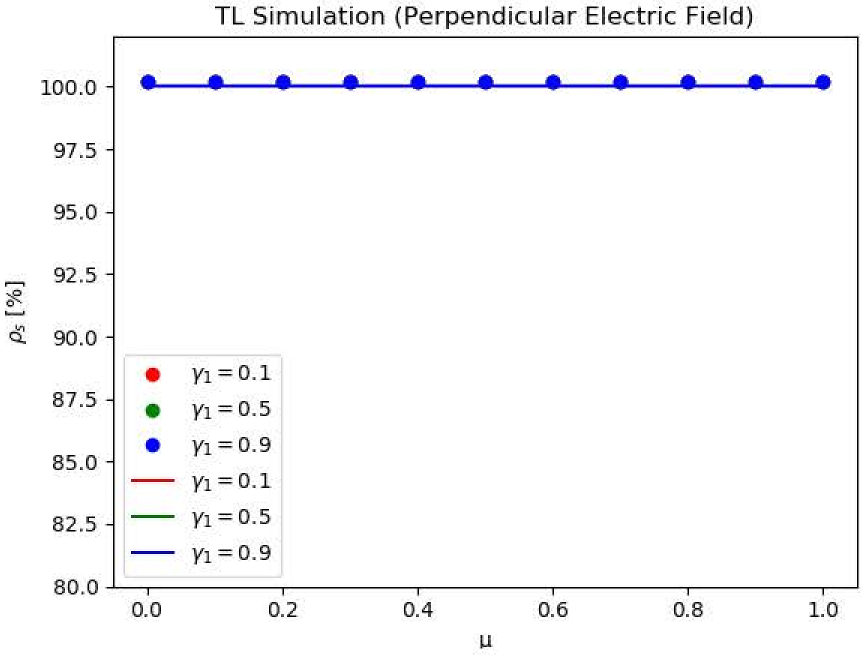

Figure 3.10: The simulated μ response (points) overlaid with the model-predicted μ response (curves) for CSAE-TL. Here, particles assemble under a perpendicular field encouraging particle attachment ($\gamma_1 \neq 0$, $\gamma_2 = 0$). The red points and curve for $\gamma_1 = 0.1$ and the green points and curve for $\gamma_1 = 0.5$ are hidden under the blue points and curve representing $\gamma_1 = 0.9$.

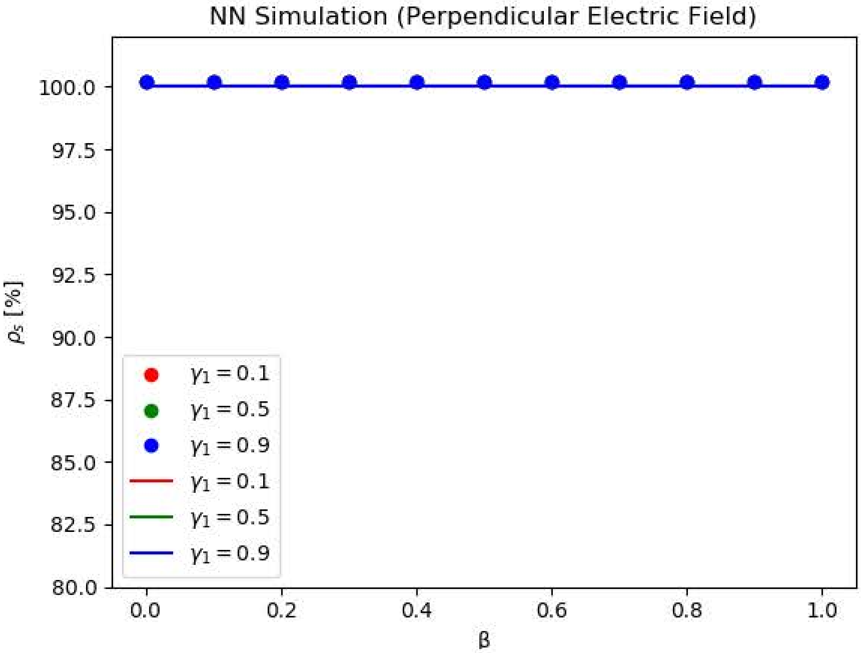

Figure 3.11: The simulated β response (points) overlaid with the model-predicted β response (curves) for CSAE-NN. Here, particles assemble under a perpendicular field encouraging particle attachment ($\gamma_1 \neq 0$, $\gamma_2 = 0$). The red points and curve for $\gamma_1 = 0.1$ and the green points and curve for $\gamma_1 = 0.5$ are hidden under the blue points and curve representing $\gamma_1 = 0.9$.

3.3 Parallel Electric Fields

The other natural orientation for our applied electric field is parallel to the surface of the slide. While the perpendicular orientation simply altered the evaporation term by forcing us to consider particles being driven into and away from the slide, the parallel orientation instead forces us to consider diffusion (i.e., the translational motion of particles on the slide surface) in addition to deposition and evaporation. In this section, we consider this scenario from the perspective of mathematical models, computer simulations, and experimental production methods, building on the work found in [46].

3.3.1 DSAM for Parallel Fields

The DSAM method for electric fields parallel to the surface of the slide is almost identical to the DSAM method for electric fields perpendicular to the slide's surface. Like in the perpendicular case, as well as with ISAM, we begin by dipping a clean slide in the polycation PDDA. The now positively charged glass can attract negatively charged SiO_2 nanoparticles. Immediately before dipping in the nanoparticle suspension, we power the parallel plate capacitor, which is now oriented perpendicular to the slide's surface. This orientation creates a uniform electric field parallel to the surface of the slide (see Figure 3.12).

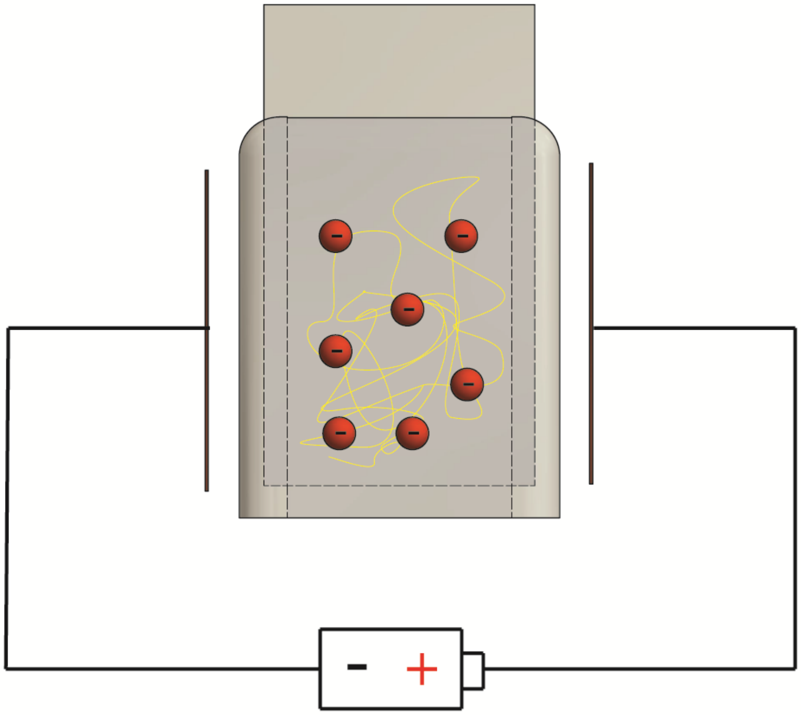

Figure 3.12: Our modification of the nanoparticle staining jar to allow for the application of a uniform electric field parallel to the slide face during self-assembly. Note that the two capacitor plates (indicated by the two vertical lines on either side of the jar) are actually adhered to the side of the jar. The separation in the image is for visual clarity only.

3.3.2 CSAE Modeling Techniques and Results

Modeling the deposition, evaporation, and diffusion of particles assembling under a parallel electric field requires reconsideration of the grid model discussed in Chapter 2.2 (see Figure 2.2). While deposition and evaporation continue to follow the same rules as described by CSAE-TL and CSAE-NN for ISAM, diffusion demands that we consider the state of particle site immediately adjacent to the site under consideration. Figure 3.13 shows why the consideration of adjacent particles is necessary. Under the influence of the electric field $\boldsymbol{E}$, each negatively charged particle has a chance of diffusing to the neighboring site *in the up-field direction.*[2] This behavior is demonstrated by the red particle, which is attempting to move up-field to the immediately adjacent site. However, diffusion is forbidden if the site in the up-field direction is occupied since the electrostatic repulsion between the two particles will prevent the diffusing particle from moving. This rule is demonstrated by the green particle, which cannot diffuse into the adjacent site in the up-field direction due to the presence of another particle. Assuming diffusion occurs with a probability λ (also scalable to the strength of the applied field using experimental data) that depends upon the strength of the

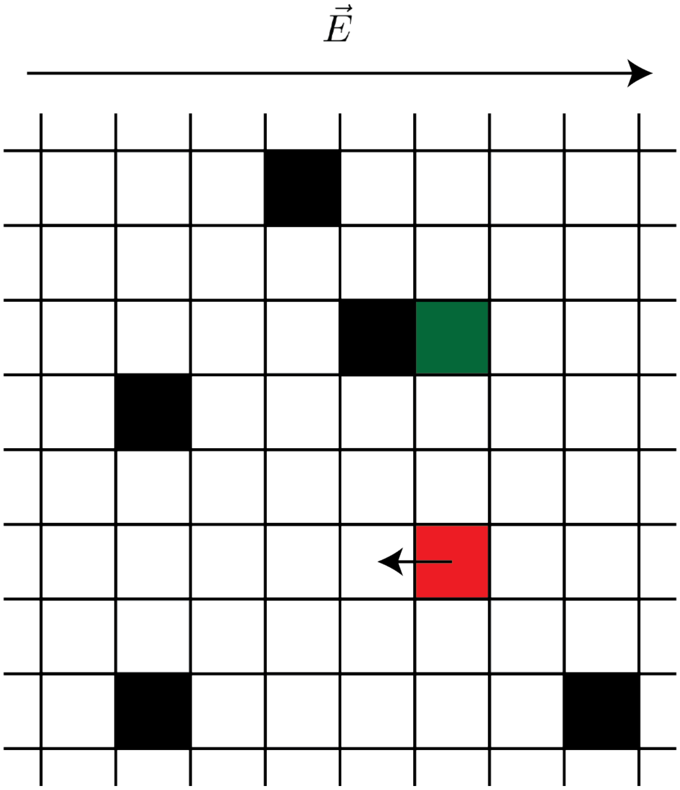

Figure 3.13: A square grid depicting the conditions governing the diffusion of negatively charged nanoparticles under a parallel electric field. The red particle may diffuse because it has no neighbor in the up-field direction. The green particle may not diffuse because it has a neighbor in the up-field direction.

[2] In the case of positively charged particles, diffusion would occur in the down-field direction.

electric field, we describe this diffusive behavior mathematically using the transition term

$$\lambda n_i(1 - n_{i\pm1}). \tag{3.20}$$

Like in our other transition terms, n_i represents the transitioning site. Thus, diffusion can only occur if $n_i = 1$, which indicates that the site is occupied. $n_{i\pm1}$ represents the immediately adjacent site in the up-field direction, with the ± sign indicating that the exact direction depends upon the orientation of the field. Thus, if the immediately adjacent site is occupied ($n_{i\pm1} = 1$) the diffusion term goes to 0 is and diffusion cannot occur. Conversely, if the immediately adjacent site is unoccupied ($n_{i\pm1} = 0$) the diffusion term goes to λ and diffusion is permitted.

Adding our diffusion term to the transition rate for the CSAE-TL model yields

$$c_{TL}(n_i \to (1 - n_i)) = n_i\gamma + \mu(1 - n_i)\left(1 - \frac{\sum_{i=1}^{n} n_i}{N}\right) + \lambda n_i(1 - n_{i\pm1}). \tag{3.21}$$

This transition rate can be expressed as the partial differential equation

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\gamma\langle n_i\rangle + \mu\left\langle(1 - n_i)\left(1 - \frac{\sum_{i=1}^{n} n_i}{N}\right)\right\rangle + \lambda\langle n_i\rangle(1 - \langle n_{i\pm1}\rangle), \tag{3.22}$$

which under the mean field approximation becomes

$$\frac{\partial\rho}{\partial t} = -\gamma\rho + \mu(1 - \rho)^2 + \lambda(\rho - \rho^2). \tag{3.23}$$

Equation (3.23) represents the rate equation for the CSAE-TL model under parallel electric fields. It suggests a steady state particle coverage density (ρ_s) of

$$\rho_s = \mp\frac{\sqrt{\gamma^2 - 2\gamma\lambda + 4\gamma\mu + \lambda^2} \pm \gamma \mp \lambda \pm 2\mu}{2\lambda - 2\mu} \tag{3.24}$$

when $\lambda \neq \mu$ and

$$\rho_s = \frac{\mu}{\gamma + \mu} \tag{3.25}$$

when $\lambda = \mu$.

If we add our diffusion term to the transition rate for the CSAE-NN model, we find

$$c_{NN}(n_i \to (1 - n_i)) = n_i\gamma + (1 - n_i)\alpha\beta^{\eta} + \lambda n_i(1 - n_{i\pm1}). \tag{3.26}$$

As a partial differential equation, this transition rate becomes

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\gamma\langle n_i\rangle + \langle(1 - n_i)\alpha\beta^{\eta}\rangle + \lambda\langle n_i\rangle(1 - \langle n_{i\pm1}\rangle). \tag{3.27}$$

Under the mean field approximation, Equation (3.27) becomes

$$\frac{\partial\rho}{\partial t} = -\gamma\rho + (1 - \rho)\alpha\beta^{z\rho} + \lambda(\rho - \rho^2), \tag{3.28}$$

the rate equation for CSAE-NN under parallel electric fields. Once again, the equation for the steady state particle coverage rate (ρ_s) is transcendental; however, we can approximate a value for ρ_s by performing a Taylor series expansion about $\beta = 1$. Using this method we find

$$\rho_s = \rho_s(\beta = 1) - (1-\beta)\left[4\frac{\alpha\rho_s(\beta=1)}{\alpha+\gamma+\lambda\rho_s(\beta=1)-\lambda}\left(1-\frac{\alpha}{\alpha+\gamma+\lambda\rho_s(\beta=1)-\lambda}\right)\right], \tag{3.29}$$

where

$$\rho_s(\beta=1) = \frac{-(\gamma+\alpha-\lambda) \pm \sqrt{(\gamma+\alpha-\lambda)^2 + 4\lambda\alpha}}{2\lambda}. \tag{3.30}$$

`odeint` can again be used to solve both Equation (3.23) and (3.28). In Figures 3.14 and 3.15, we show the change in steady state particle coverage density as μ (for CSAE-TL) and β (for CSAE-NN) at several λ values. Steady state coverage density once again increases as μ and β increase, mirroring physical reality. Raising the diffusion coefficient λ, which corresponds to increasing the strength of the parallel electric field, seems to suggest elevated ρ_s values in both models. Nevertheless, the ρ_s curve maintains its overall shape no matter the λ value used. Like our models for assembly under no electric field field and perpendicular electric fields, the linear behavior of ρ_s under the CSAE-NN model (see Figure 3.15) is a result of our Taylor series approximation, which we discuss further in Chapter 4.

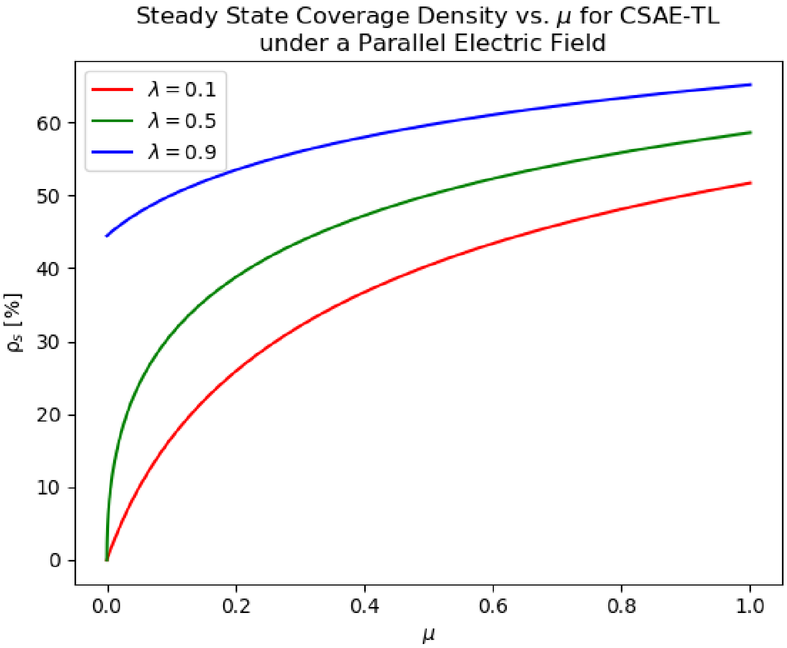

Figure 3.14: The μ response of steady state particle coverage density calculated using our CSAE-TL model. Here, particles assemble under a parallel field . Notice that a higher μ value corresponds to a higher ρ_s value. Also notice that changing γ alters the value of ρ_s but not the shape of the response curve.

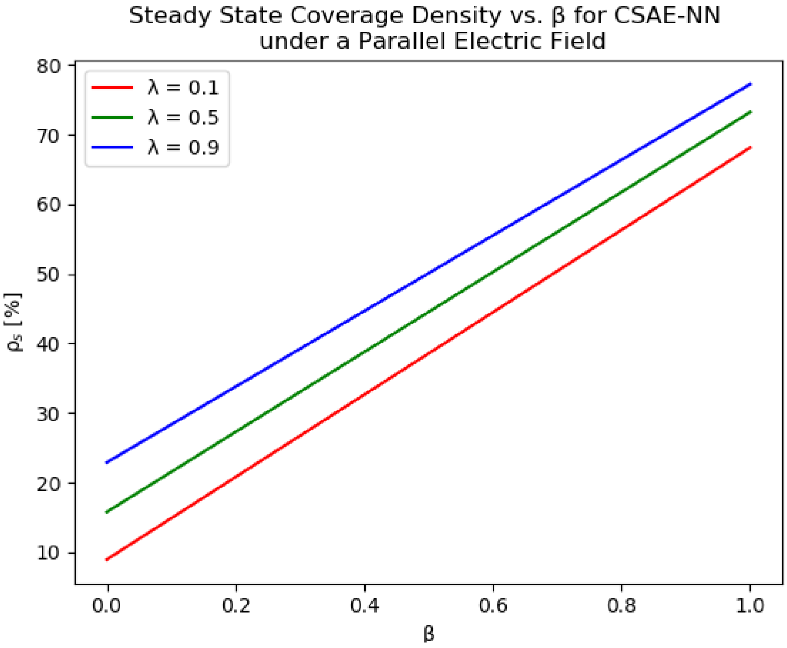

Figure 3.15: The β response of steady state particle coverage density calculated using our CSAE-NN model. Here, particles assemble under a parallel field . Notice that a higher β value corresponds to a higher ρ_s value. Also notice that changing γ alters the value of ρ_s but not the shape of the response curve.

3.3.3 Simulation Techniques and Results

Like in our simulations of nanoparticle assembly under no electric fields and perpendicular electric fields, we use a Monte Carlo-based method to simulate nanoparticle assembly under parallel electric fields. The algorithm contains one major difference, however. If the comparison of our test value P and random value R leaves a particular cell unchanged (i.e., if neither deposition nor evaporation occurs), the site can still change state via diffusion. If the unchanged cell is occupied, the program checks the cell in the up-field direction to determine its state. If this neighboring cell is occupied, no diffusion occurs. If it is not occupied, the program determines whether diffusion occurs by comparing the random value R to our diffusion probability λ. If $R \leq \lambda$, diffusion occurs, and the program changes the states of the unchanged cell and the neighboring cell in the up-field direction. If the unchanged cell is unoccupied, the program checks the cell in the down-field direction to determine its state. If this neighboring cell is unoccupied, no diffusion occurs. If it is occupied, the program determines whether diffusion occurs by comparing the random value R to our diffusion probability λ. If $R \leq \lambda$, diffusion occurs, and the program changes the states of the unchanged cell and the neighboring cell in the down-field direction.[3]

Figure 3.16 and 3.17 show steady state particle coverage densities for simulated particle assembly following the CSAE-TL and CSAE-NN transition rates using a variety of λ values. Each simulation was completed on a 100 × 100 grid over 1000 time steps. In Chapter 4, we compare

[3] Note that this description is appropriate for a simulation of negatively charged particles. All instances of "up-field" and "down-field" would switch for a simulation of positively charged particles.

these results to our modeling results (Figures 3.14 and 3.15).

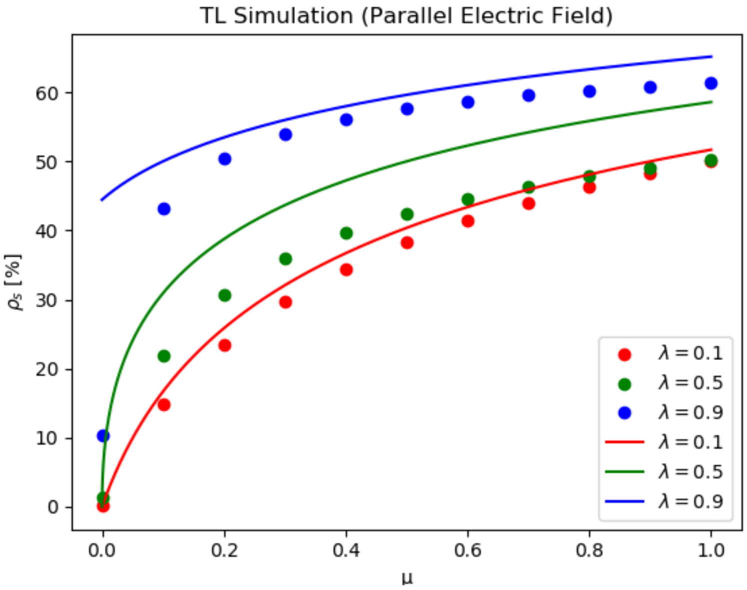

Figure 3.16: The simulated μ response (points) overlaid with the model-predicted μ response (curves) for CSAE-TL. Here, particles assemble under a parallel field.

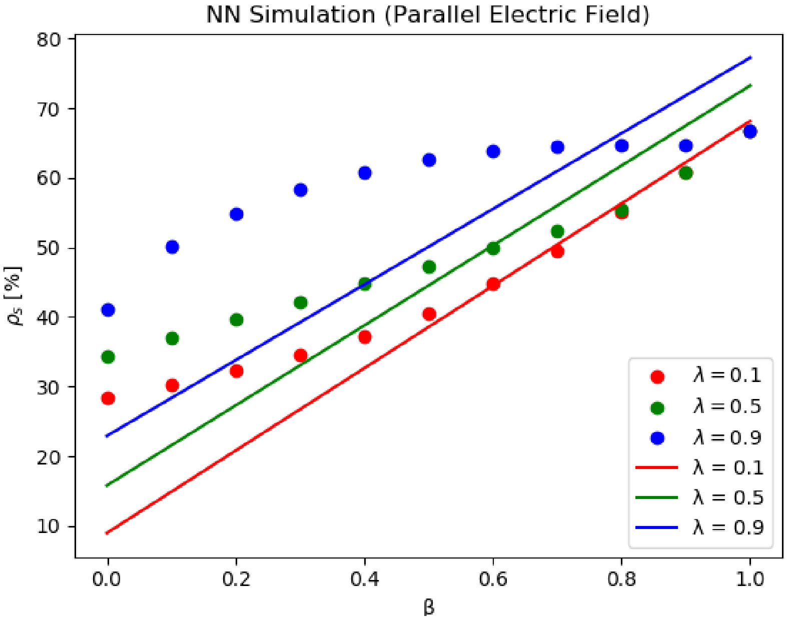

Figure 3.17: The simulated β response (points) overlaid with the model-predicted β response (curves) for CSAE-NN. Here, particles assemble under a parallel field.

3.4 Oscillating Electric Fields

3.4.1 Overview of Oscillating Fields

Thus far, our treatment of electric field-driven DSAM has been limited to the use of constant fields produced by a capacitor attached to a direct current (DC) power source. However, assembling particles can also experience oscillating fields, easily produced by attaching a capacitor to an alternating current (AC) power source. For brevity and to remind the reader of their experimental origins, we refer to constant fields as "DC fields" and oscillating fields as "AC fields" throughout the following sections.

We touch on the consequences of assembly under AC fields briefly in [53]. In this case, the strength and orientation of the field changes over time. Thus, assembling nanoparticles will oscillate back and forth as the field changes direction. We can apply AC fields to a system under assembly in any orientation. Furthermore, we can also superimpose AC fields and DC fields. For the sake of simplicity, our mathematical and experimental treatment focuses on four main scenarios:

1. Perpendicular AC Field

2. Parallel AC Field

3. Perpendicular AC Field; Parallel DC Field

4. Parallel AC Field; Perpendicular DC Field

Scenarios (3) and (4) are especially important because there is reason to believe that the initial kinetic energy imparted to assembling nanoparticles by a small AC field can aid in a DSAM process that is primarily driven by a DC field.

3.4.2 CSAE Modeling Techniques and Results

Like with our CSAE models for DSAM under DC fields, we produce models for our selected AC field cases by constructing transition rates, converting them to partial differential equations, and applying the mean field approximation. This provides differential equations describing the particle coverage density ρ over time. We can solve these equations numerically for all times. In the steady state, we can also use algebraic manipulation (CSAE-TL) or Taylor series expansions (CSAE-NN) to solve for the approximate steady state particle coverage density ρ_s without the use of numerical solving tools.

To introduce an oscillating field, our transition rates must include a new coefficient, equivalent in nature to the evaporation (γ), deposition (μ, α, and β), and diffusion (λ) coefficients we have already discussed. As a reminder, each coefficient represents the probability that a nanoparticle will attach/detach from location on a grid representing the slide surface. However, this AC-field coefficient must be able to vary its strength in the same way that the voltage of the AC power source varies. For a sinusoidal power source, we have

$$\chi\cos(\omega t + \phi), \tag{3.31}$$

where χ is the maximum value of the AC-field coefficient (scaled to the amplitude of the power source), ω is the angular frequency of the sinusoidal source, and ϕ is the phase shift of the sinusoidal source. When the AC-field is not oriented perfectly perpendicular or parallel to the slide surface, this coefficient resolves into two components, which we call $\chi_\gamma\cos(\omega t+\phi)$ for the perpendicular component and $\chi_\lambda\cos(\omega t+\phi)$ for the parallel component.[4] For clarity throughout our derivations, we use χ_γ and χ_λ explicitly.

For scenario (1), in which we only apply a perpendicular AC field, the transition rate for CSAE-NN[5] is

$$c(n_i \to (1-n_i)) = n_i\gamma + (1-n_i)\alpha\beta^{\eta} + n_i\chi_\gamma\cos(\omega t+\phi). \tag{3.32}$$

This equation assumes that the field only oscillates the state of a site that is already occupied. More complicated models could also consider oscillations of non-occupied sites. As a differential equation, this transition rate becomes

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\langle n_i\rangle\gamma + \langle(1-n_i)\alpha\beta^{\eta}\rangle + \langle n_i\rangle\chi_\gamma\cos(\omega t+\phi). \tag{3.33}$$

The mean field approximation then produces

$$\frac{\partial\rho}{\partial t} = -\rho\gamma + (1-\rho)\alpha\beta^{z\rho} + \rho\chi_\gamma\cos(\omega t+\phi). \tag{3.34}$$

Figure 3.18 shows a numerical solution for this equation across a wide range of times. We see that

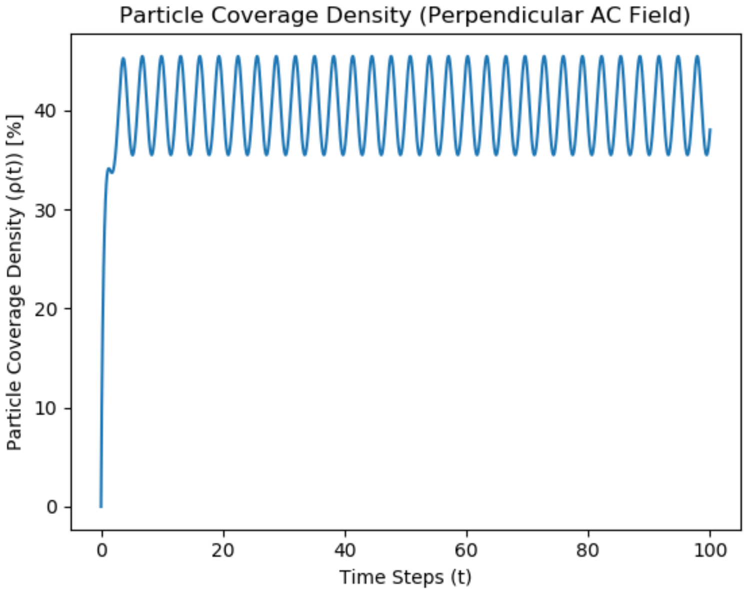

Figure 3.18: Numerical solution to the CSAE-NN model for particles assembling under a perpendicular AC field. ($\gamma=\beta=0.5$, $\alpha=1$, $z=4$, $\chi_\gamma=0.3$.)

[4] We select γ for the perpendicular field case because the coefficient behaves like a time-varying evaporation coefficient and λ for the parallel field case because the coefficient behaves like a time-varying diffusion coefficient.

[5] The CSAE-TL model for this case is constructed by simply replacing the NN deposition term with the TL deposition term (see Equation (2.1)).

the system rapidly approaches steady state, just like in the no-field and DC field cases. However, because of the presence of the perpendicular AC field, the particle coverage density oscillates sinusoidally around an average steady state particle coverage density (in this case, $\langle \rho_s \rangle \approx 40\%$). These are the results we expect, for, as the AC field oscillates, particles alternate between being driven into and away from the face of the slide, altering the coverage density periodically even in the steady state regime.

Scenario (2) describes the application of a parallel AC field. The CSAE-NN transition rate is

$$c(n_i \to (1-n_i)) = n_i\gamma + (1-n_i)\alpha\beta^{\eta} + n_i(1-n_{i-1})(1-n_{i+1})\chi_\lambda \cos(\omega t+\phi). \tag{3.35}$$

Like in the perpendicular case, we only permit oscillations of sites that are occupied. Furthermore, the two neighboring sites (n_{i-1} and n_{i+1}) must also be unoccupied. If n_{i-1} and n_{i+1} were occupied, the electrostatic repulsion produced by the particles in these sites would inhibit oscillation. As a differential equation, this transition rate becomes

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\langle n_i\rangle\gamma + \langle(1-n_i)\alpha\beta^{\eta}\rangle + \langle n_i\rangle(1-\langle n_{i-1}\rangle)(1-\langle n_{i+1}\rangle)\chi_\lambda \cos(\omega t+\phi). \tag{3.36}$$

After applying the mean field approximation, we have

$$\frac{\partial\rho}{\partial t} = -\rho\gamma + (1-\rho)\alpha\beta^{z\rho} + \rho(1-\rho)^2\chi_\lambda \cos(\omega t+\phi). \tag{3.37}$$

Figure 3.19 is a representative numerical solution of this equation. These results are similar to the results we obtained for the perpendicular AC field: the system rapidly approaches steady state,

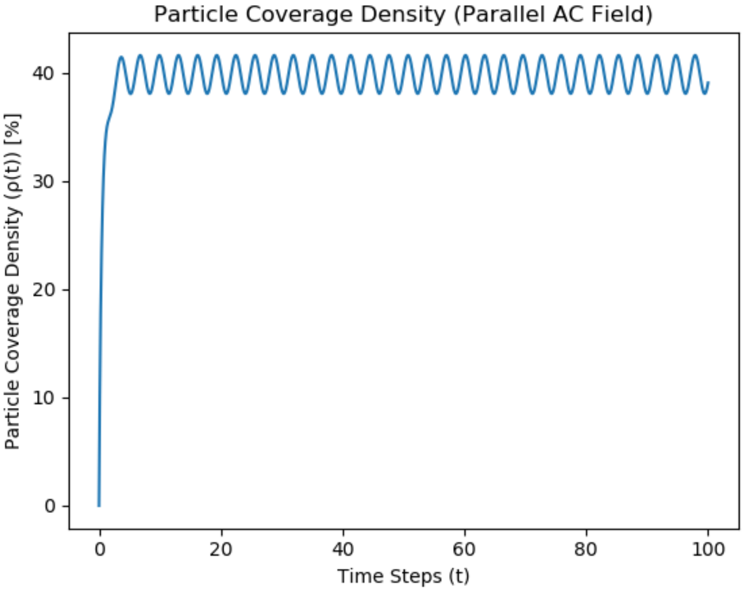

Figure 3.19: Numerical solution to the CSAE-NN model for particles assembling under a parallel AC field. ($\gamma = \beta = 0.5$, $\alpha = 1$, $z = 4$, $\chi_\lambda = 0.3$.)

at which point the AC field forces the system to oscillate about an average steady state particle coverage density (in this case, like in Figure 3.18, $\langle\rho_s\rangle \approx 40\%$). The main distinction between the parallel and perpendicular case is the amplitude of the coverage density oscillations. Even when we examine scenarios where $\chi_\gamma = \chi_\lambda$, the amplitude is always smaller in the parallel field case. This is because, under a parallel field, particle coverage density can only increase or decrease as particles diffuse onto or off the slide at its edges. Under a perpendicular field, particles can be driven onto or away from the slide at any point, greatly increasing the degree to which the coverage density can change in steady state.

Scenario (3) includes both a perpendicular AC field and a parallel DC field. We construct the transition rate as follows:

$$c(n_i \to (1-n_i)) = n_i\gamma + (1-n_i)\alpha\beta^{\eta} + n_i(1-n_{i\pm1})\lambda + n_i\chi_\gamma\cos(\omega t+\phi). \tag{3.38}$$

Here, $n_i(1-n_{i\pm1})\lambda$ represents the parallel DC field, which is only active for an occupied site with an unoccupied neighbor in the up-field direction. $n_i\chi_\gamma\cos(\omega t+\phi)$ represents the AC field and is only active for an occupied site. As a differential equation, we have

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\langle n_i\rangle\gamma + \langle(1-n_i)\alpha\beta^{\eta}\rangle + \langle n_i\rangle(1-\langle n_{i\pm1}\rangle)\lambda + \langle n_i\rangle\chi_\gamma\cos(\omega t+\phi). \tag{3.39}$$

The mean field approximation then gives us

$$\frac{\partial\rho}{\partial t} = -\rho\gamma + (1-\rho)\alpha\beta^{z\rho} + \rho(1-\rho)\lambda + \rho\chi_\gamma\cos(\omega t+\phi). \tag{3.40}$$

Figure 3.20 is a representative numerical solution of this model. These results are very similar

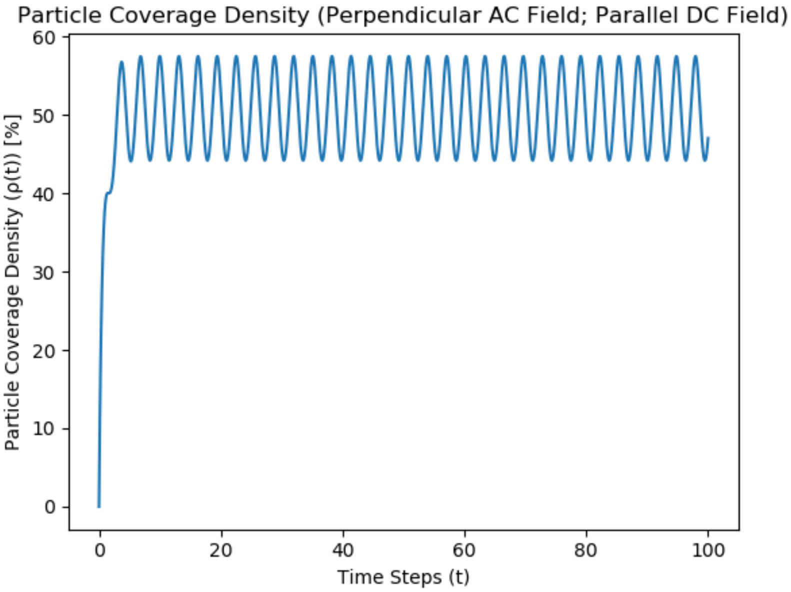

Figure 3.20: Numerical solution to the CSAE-NN model for particles assembling under a perpendicular AC field and a parallel DC field. ($\gamma = \beta = \lambda = 0.5$, $\alpha = 1$, $z = 4$, $\chi_\gamma = 0.3$.)

to Scenario (1) (see Figure 3.18), except that the presence of the parallel DC field increases the average steady state particle coverage density value.

Finally, scenario (4) proposes a parallel AC field superimposed with a perpendicular DC field. The required transition rate is

$$c(n_i \to (1-n_i)) = (1-n_i)\gamma_1 + n_i\gamma_2 + (1-n_i)\alpha\beta^{\eta} + n_i(1-n_{i-1})(1-n_{i+1})\chi_\lambda \text{cos}(\omega t + \phi). \tag{3.41}$$

The terms $(1-n_i)\gamma_1 + n_i\gamma_2$ describe the perpendicular DC field, with the orientation determined by whether γ_1 or γ_2 is non-zero. The term $n_i(1-n_{i-1})(1-n_{i+1})\chi_\lambda \text{cos}(\omega t+\phi)$ describes the parallel AC field. As a differential equation, we have

$$\frac{\partial\langle n_i\rangle}{\partial t} = (1-\langle n_i\rangle)\gamma_1 - \langle n_i\rangle\gamma_2 + \langle(1-n_i)\alpha\beta^{\eta}\rangle + \langle n_i\rangle(1-\langle n_{i-1}\rangle)(1-\langle n_{i+1}\rangle)\chi_\lambda \text{cos}(\omega t+\phi). \tag{3.42}$$

The mean field approximation then produces

$$\frac{\partial\rho}{\partial t} = (1-\rho)\gamma_1 - \rho\gamma_2 + (1-\rho)\alpha\beta^{z\rho} + \rho(1-\rho)^2\chi_\lambda \text{cos}(\omega t+\phi). \tag{3.43}$$

Figure 3.21 and 3.22 are numerical solutions for this equation when $\gamma_1 = 0$ and $\gamma_2 = 0$, respectively. Figure 3.21 shows behavior identical to Figure 3.19 since the perpendicular DC field, which is driving particles away from the slide, has the same effect as evaporation. However, Figure 3.22 shows that, when a parallel AC field is combined with a perpendicular DC field (driving particles into the slide) that is strong enough to achieve $\rho_s = 100\%$, oscillations about $\langle\rho_s\rangle$ fail to appear.

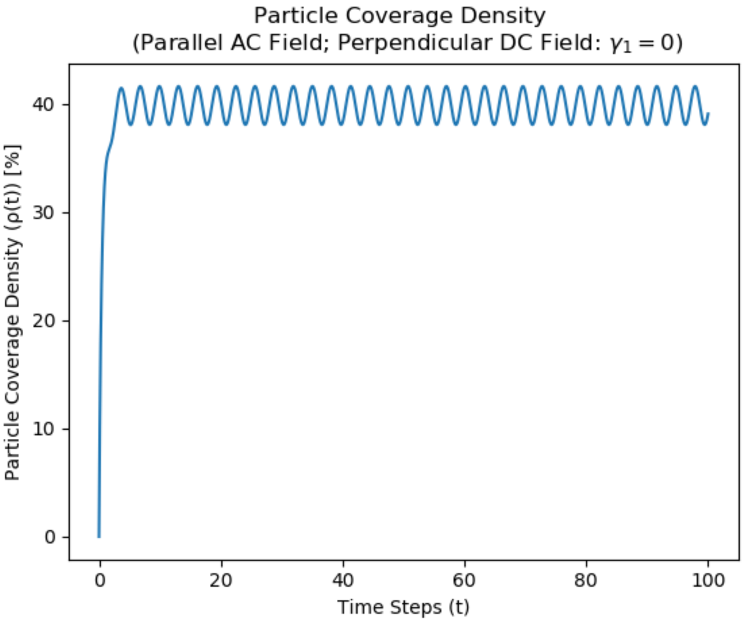

Figure 3.21: Numerical solution to the CSAE-NN model for particles assembling under a parallel AC field and a perpendicular DC field which encourages particle detachment. ($\gamma_1 = 0$, $\gamma_2 = \beta = 0.5$, $\alpha = 1$, $z = 4$, $\chi_\lambda = 0.3$.)

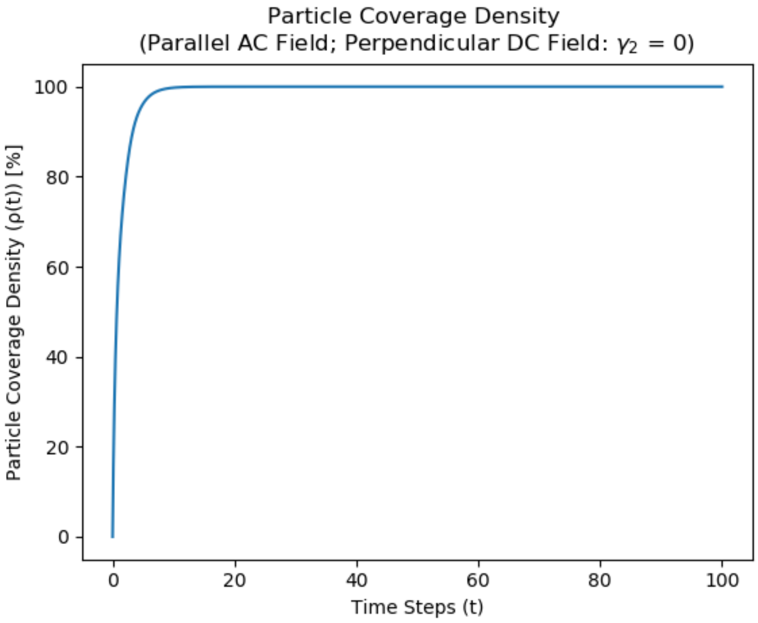

Figure 3.22: Numerical solution to the CSAE-NN model for particles assembling under a parallel AC field and a perpendicular DC field which encourages particle attachment. ($\gamma_2 = 0$, $\gamma_1 = \beta = 0.5$, $\alpha = 1$, $z = 4$, $\chi_\lambda = 0.3$.)

3.5 Experimental Techniques

Thus far, our efforts to produce bilayers under the influence of perpendicular and parallel electric fields in the laboratory have focused on experiment design. In general, we use the same steps outlined in Figure 2.1 and described in detail in Chapter 2.1: (1) we dip a clean slide into a suspension of PDDA, (2) we rinse the PDDA-coated slide in DI water, (3) we dip the rinsed slide into a suspension of SiO_2 nanoparticles, (4) we rinse again in DI water, and (5) we dry under a gentle flow of N_2 gas. However, for the application of an electric field during deposition, we must modify the staining jar holding the nanoparticles. By placing a sheet of copper tape on the two faces parallel to the slide, we form a capacitor which delivers a uniform electric field perpendicular to the slide face. By placing a sheet of copper tape on the two faces perpendicular to the slide, we form a capacitor which delivers a uniform electric field parallel to the slide face. This is shown schematically in Figures 3.3 and 3.12. Figure 3.23 shows one of our laboratory beakers, which contains copper tape sheets on all sides. Thus, we can select a perpendicular or parallel field arrangement by simply attaching our electrical connections to the appropriate pair of sides.

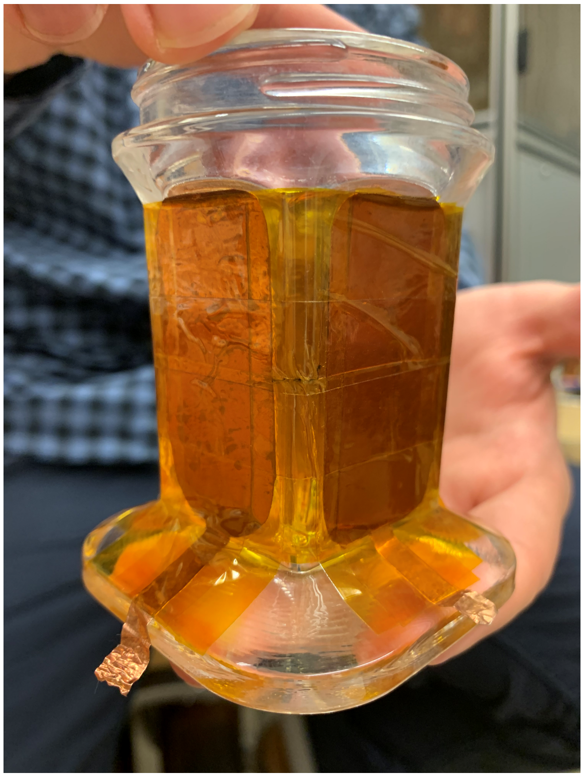

Figure 3.23: Our modified staining jar for the application of electric fields parallel and perpendicular to the surface of a slide during nanoparticle assembly. The capacitor plates are constructed from copper tape. Insulating tape prevents sparking between plates when two fields are applied simultaneously.

The two strips of tape form a capacitor when attached to a DC power source. Just before we dip the slide into the SiO_2 suspension, we activate the DC power source, which we previously set to the desired voltage level. Activating the DC power source just before deposition begins is important, for, if the capacitor is charged long before deposition, a gradient of SiO_2 nanoparticles can form within the suspension. Such a gradient would obscure our results.

Currently, we are investigating the electric field strength needed to affect the particles and produce DSAM. As we discuss in Chapter 3.1, the strength of the electric field is given by

$$E = \frac{V}{d}. \tag{3.44}$$

Thus, we can increase the electric field by either increasing the voltage V applied to the capacitor or decreasing the separation d between the capacitor plates. We have made preparations to take both approaches. To increase the applied voltage, we have acquired a BT-GP-10N30 power supply from Advance Energy, which produces DC voltages between 0 and 10,000 V. With our standard staining jar, which has a width of 39.3 mm, this supply can produce a uniform field between 0 and 254 kV/m. To decrease the plate separation, we have produced several specialty staining jars using a 3D printer.

Applying AC electric fields during the assembly process is identical to applying a DC field. However, instead of a DC power source, we connect a function generator (SRS DS335) to the capacitor plates. Most commercially available function generators do not produce a peak voltage with enough strength to encourage particle oscillations. This problem can be rectified by introducing an amplifier (Trek 2205), which increases the peak voltage of the function generator without altering the angular frequency and shape of the signal. Like with deposition under DC fields, our efforts to produce nanoparticles bilayers under the influence of an AC fields is currently in the experiment design stage.

Chapter 4

Discussion: Comparing CSAE Models and Simulated Data

The utility of our CSAE models depends upon their ability to predict the coverage densities of simulated and experimentally produced bilayer samples. Because our experimental work is ongoing, we consider here only the fit of our models to simulated data. Furthermore, we focus primarily on no-field and DC-field cases, as, to date, these have received the most attention. We conduct our comparison using a graphical approach which relies upon residuals [54]. For any variable x (in our case, μ for CSAE-TL models and β for CSAE-NN models), the residual $\boldsymbol{e_i}$ is simply a vector containing the differences between the simulated/experimental values $\boldsymbol{y_i}$ and the values produced by the model $f(\boldsymbol{x_i})$:

$$\boldsymbol{e_i} = \boldsymbol{y_i} - f(\boldsymbol{x_i}). \tag{4.1}$$

By plotting the residual vector against $\boldsymbol{x_i}$ on a scatter plot, we can examine the resulting shape to determine the closeness of the model fit.

Figures 4.1 - 4.8 are the residual scatter plots for our models. A perfect model should produce a residual of zero for every data point comparison. The residual scatter of a strong, predictive model should demonstrate no structure; rather, the points should be distributed stochastically around zero. Finally, the residual of a poor model shows a non-stochastic mathematical structure. This structure indicates the model has failed to capture some aspect of the examined variable's behavior. Visual inspection of the residual scatter plots of our models shows that the CSAE-TL models for no electric field and perpendicular electric fields have strong predictive capabilities across all values of μ. There is a slight positive trend (i.e., stochastic behavior appears to be centered slightly above zero); however, the order of magnitude (10^{-4} - 10^{-3}) makes this discrepancy insignificant. Our CSAE-TL model for parallel fields and CSAE-NN models demonstrate a distinct structure. This suggests that the mean field approximation has failed to capture every aspect of the behavior of μ (for CSAE-TL) of β (for CSAE-NN). We expect this behavior because, according to [55], mean field theory is only capable of capturing the qualitative behavior of a many-particle, multi-state system. A full treatment, which includes interactions between particles, is necessary to develop an quantitatively accurate model. In the case of particle assembly, analytical models of this type are impossible. Thus, we must be satisfied with models of the type we have developed, which, as shown in Figures 2.14, 2.16, 3.8 - 3.11, and 3.16 - 3.17, can predict the general trend of the data (e.g., whether it is increasing or decreasing; the response of secondary variables like γ, γ_1, γ_2, and λ; etc.). For calculations of particle coverage density that do not require high precision, this qualitative agreement suggests that our models are useful, especially when simulation techniques are not available or are prohibitively time consuming.

Further visual inspection of Figures 2.14, 2.16, 3.8 - 3.11, and 3.16 - 3.17 seems to suggest that our CSAE models conform more closely with simulated data for higher values of μ and β (i.e., for $\mu,\ \beta \geq 0.6$). To test this hypothesis, we generate new residual scatters for $\mu,\ \beta \geq 0.6$ only (Figures 4.9 - 4.16). These plots show a slight improvement in the performance of our models at higher values of μ and β. While purely stochastic behavior fails to appear, especially in the CSAE-NN models, the order of magnitude of the residuals falls. Therefore, we can confidently say that our models, while still qualitative in nature, perform better when $\mu,\ \beta \geq 0.6$.

There are a number of potential improvements that could be made to our models. First, including higher order terms in the Taylor series expansions of our CSAE-NN models has the potential to produce mathematical descriptions that better capture the response of ρ_s to β. Second, we could consider a more sophisticated version of mean field theory, such as the Bethe approximation [55]. The most accurate description of a many-particle, multi-state system allows each particle to interact with every other particle in the system. This is especially true of ISAM models since every particle experiences a different electrostatic force from every other particle in the system during assembly. Mathematical treatments that include every interaction are typically unsolvable. Simple mean field theories, such as the one we have employed, assume that each individual particle experiences an effective field produced by the other particles instead of individual interactions (see Appendix A for more detail). While this description is mathematically tenable, it eliminates a significant amount of physical information because interactions between individual particle are not considered. The Bethe approximation attempts to find a middle ground between these two approaches. Each particle interacts directly with its z nearest neighbors, forming a particle cluster. The remainder of the particles then form an effective field. While still taking advantage of the mean field's ability to produce solvable mathematics, the Bethe approximation captures more detail, which could drastically improve the predictive abilities of our models.

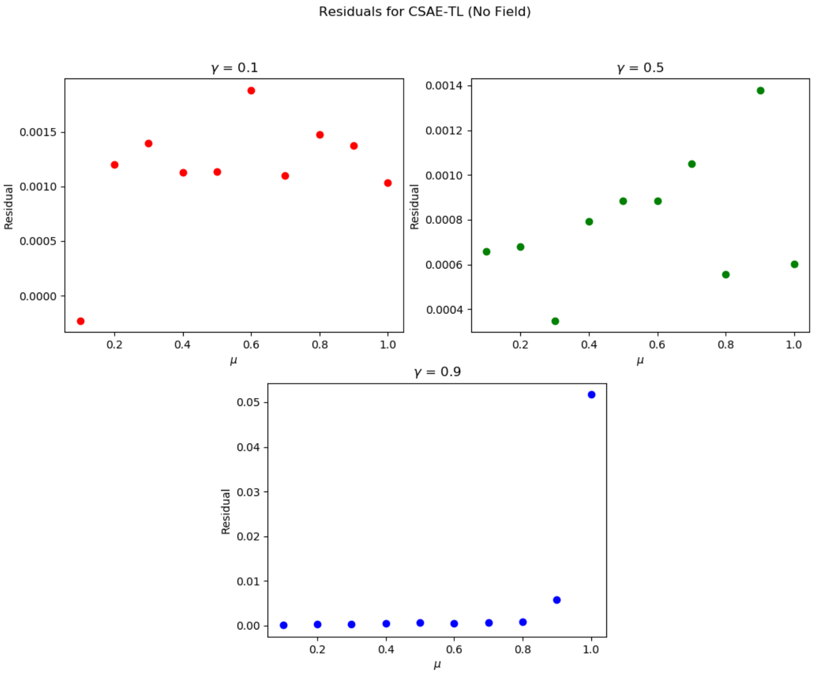

Figure 4.1: Residuals for CSAE-TL (No Field). (Red: $\gamma = 0.1$; Green: $\gamma = 0.5$; Blue: $\gamma = 0.9$.)

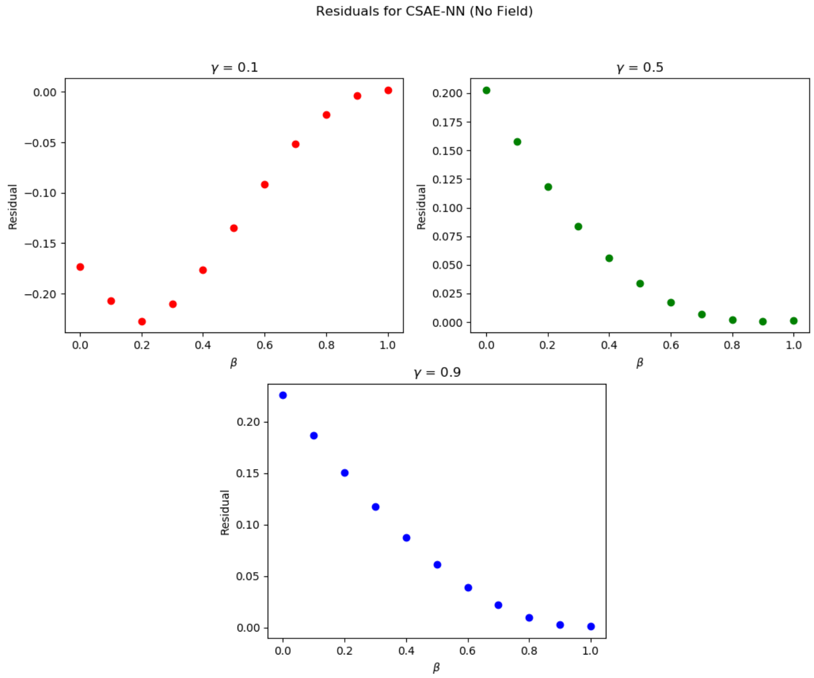

Figure 4.2: Residuals for CSAE-NN (No Field). (Red: $\gamma = 0.1$; Green: $\gamma = 0.5$; Blue: $\gamma = 0.9$.)

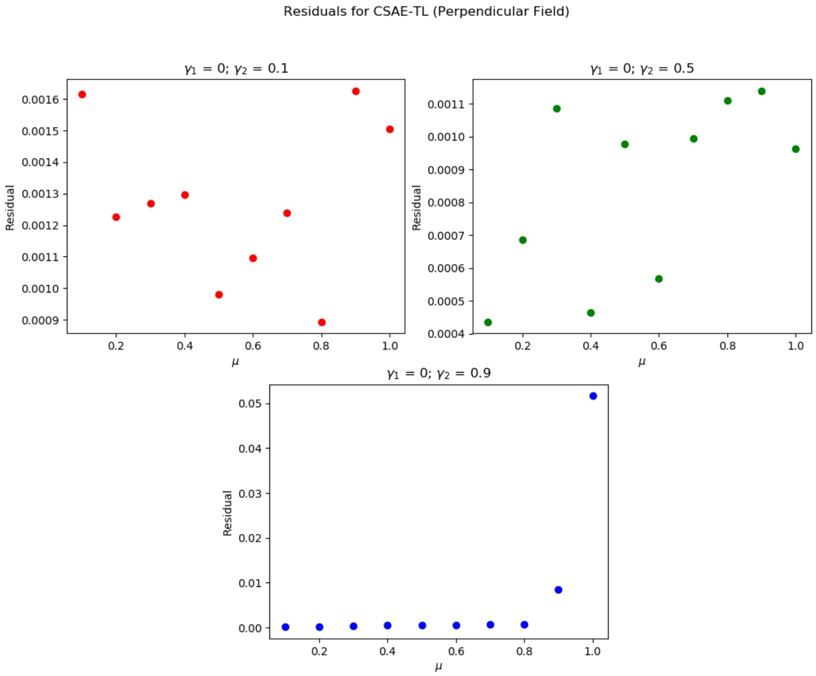

Figure 4.3: Residuals for CSAE-TL (⊥ Field). (Red: $\gamma_2 = 0.1$; Green: $\gamma_2 = 0.5$; Blue: $\gamma_2 = 0.9$.)

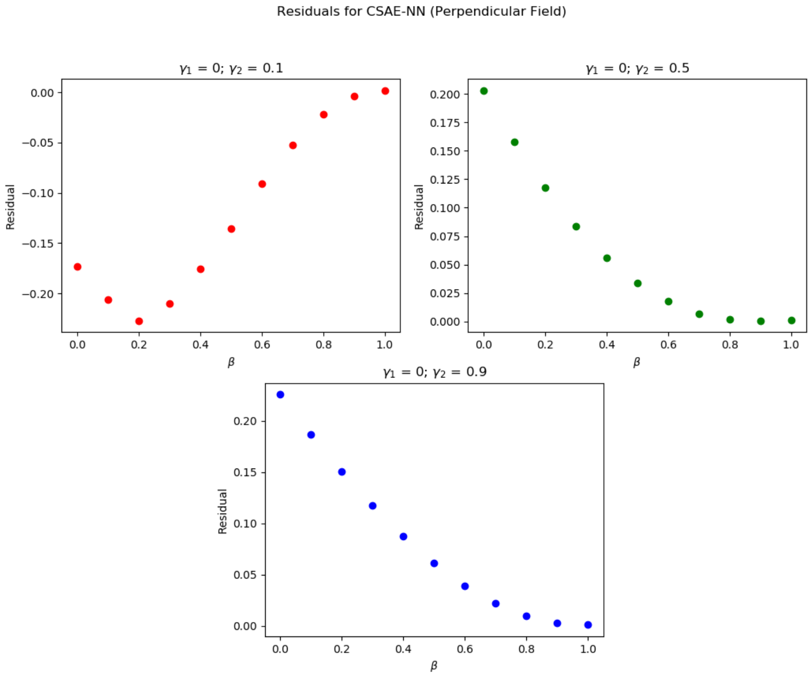

Figure 4.4: Residuals for CSAE-NN (⊥ Field). (Red: $\gamma_2 = 0.1$; Green: $\gamma_2 = 0.5$; Blue: $\gamma_2 = 0.9$.)

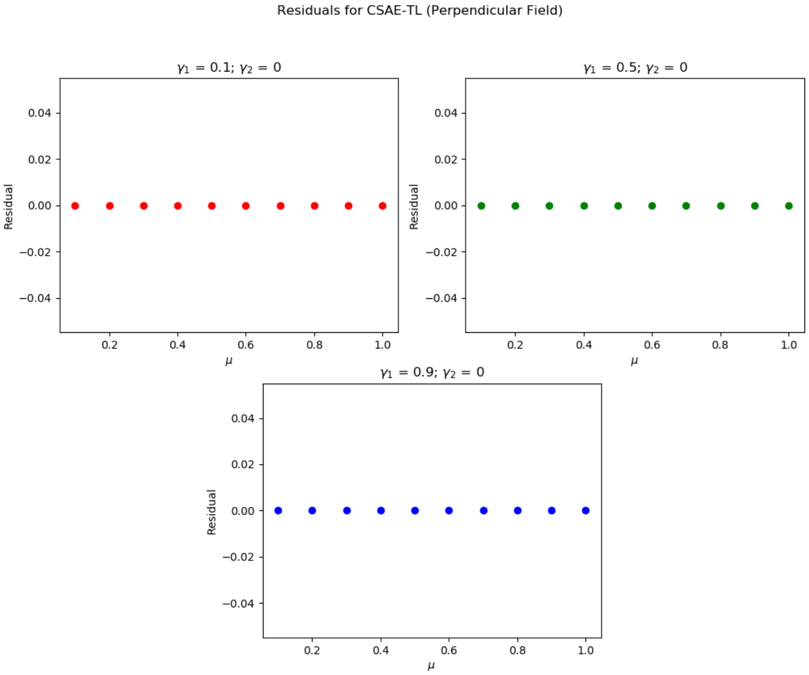

Figure 4.5: Residuals for CSAE-TL ($\perp$ Field). (Red: $\gamma_1 = 0.1$; Green: $\gamma_1 = 0.5$; Blue: $\gamma_1 = 0.9$.)

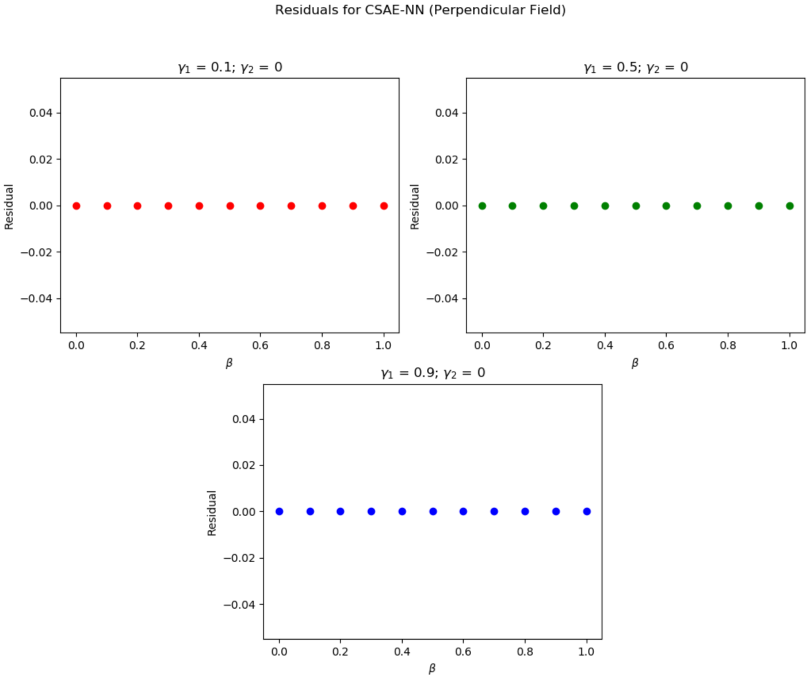

Figure 4.6: Residuals for CSAE-NN ($\perp$ Field). (Red: $\gamma_1 = 0.1$; Green: $\gamma_1 = 0.5$; Blue: $\gamma_1 = 0.9$.)

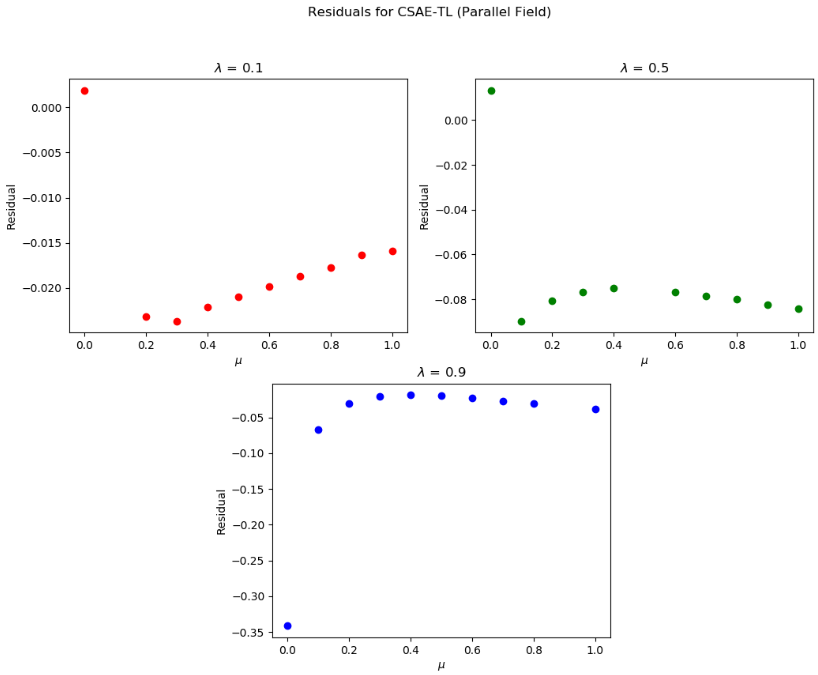

Figure 4.7: Residuals for CSAE-TL (∥ Field). (Red: $\lambda = 0.1$; Green: $\lambda = 0.5$; Blue: $\lambda = 0.9$.)

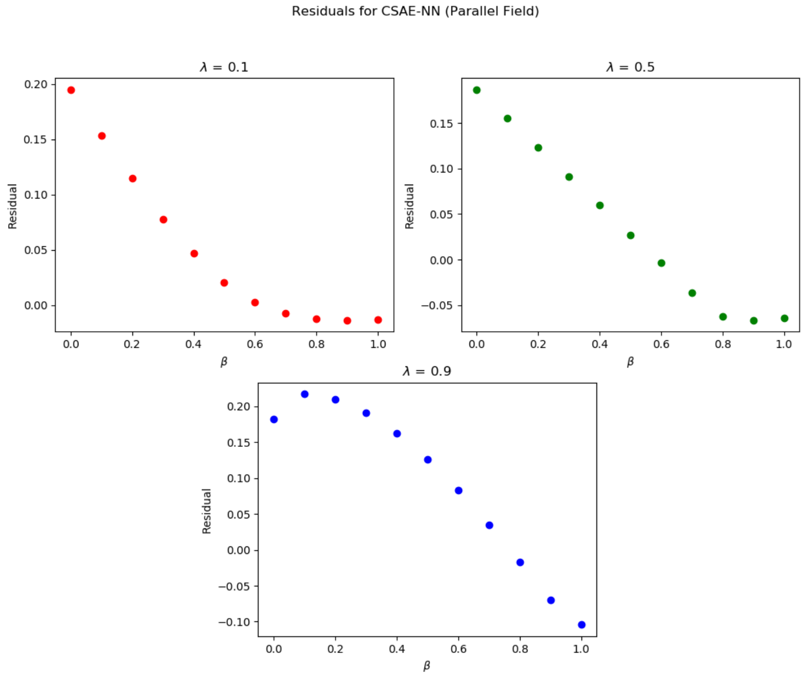

Figure 4.8: Residuals for CSAE-NN (∥ Field). (Red: $\lambda = 0.1$; Green: $\lambda = 0.5$; Blue: $\lambda = 0.9$.)

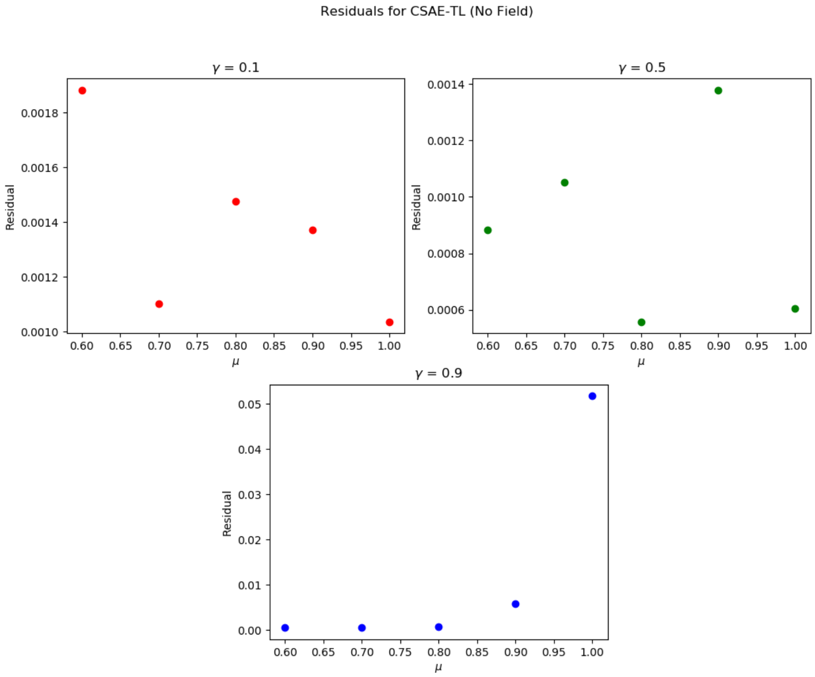

Figure 4.9: Residuals for CSAE-TL (No Field). (Red: $\gamma = 0.1$; Green: $\gamma = 0.5$; Blue: $\gamma = 0.9$.)

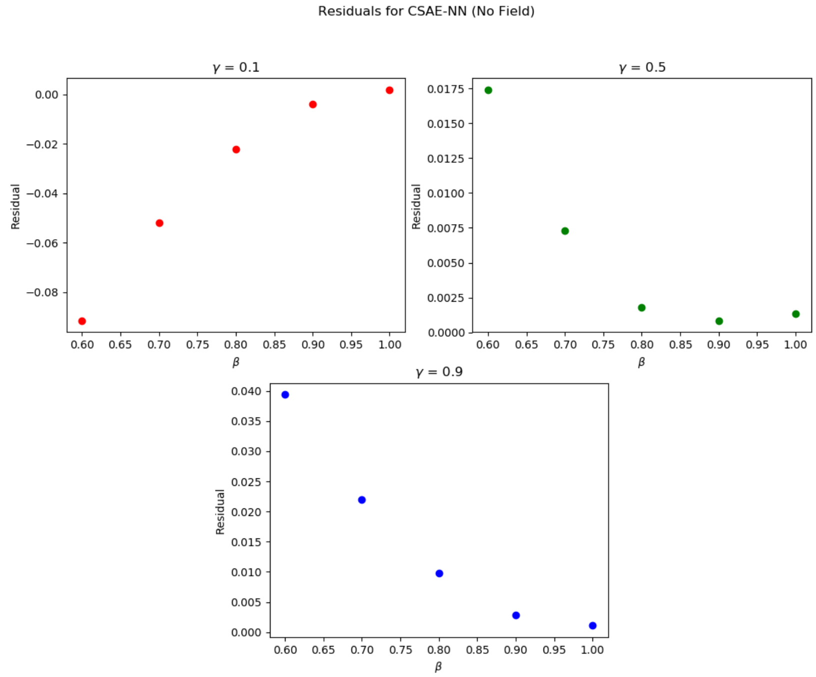

Figure 4.10: Residuals for CSAE-NN (No Field). (Red: $\gamma = 0.1$; Green: $\gamma = 0.5$; Blue: $\gamma = 0.9$.)

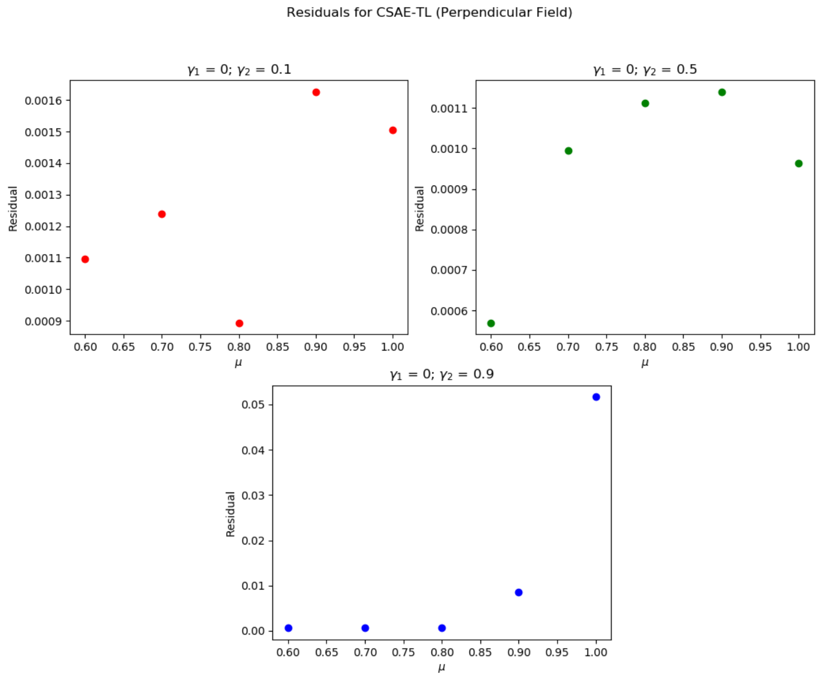

Figure 4.11: Residuals for CSAE-TL ($\perp$ Field). (Red: $\gamma_2 = 0.1$; Green: $\gamma_2 = 0.5$; Blue: $\gamma_2 = 0.9$.)

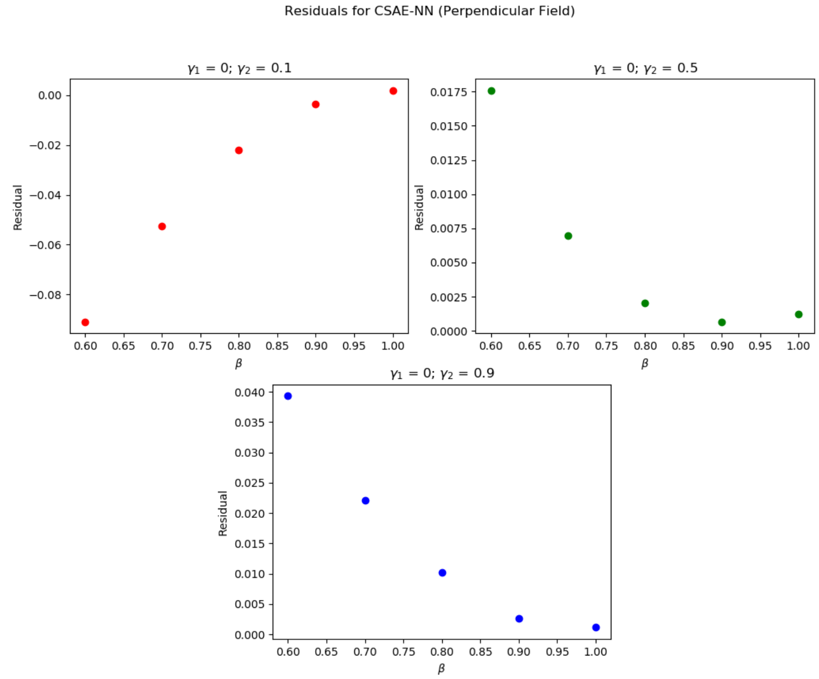

Figure 4.12: Residuals for CSAE-NN ($\perp$ Field). (Red: $\gamma_2 = 0.1$; Green: $\gamma_2 = 0.5$; Blue: $\gamma_2 = 0.9$.)

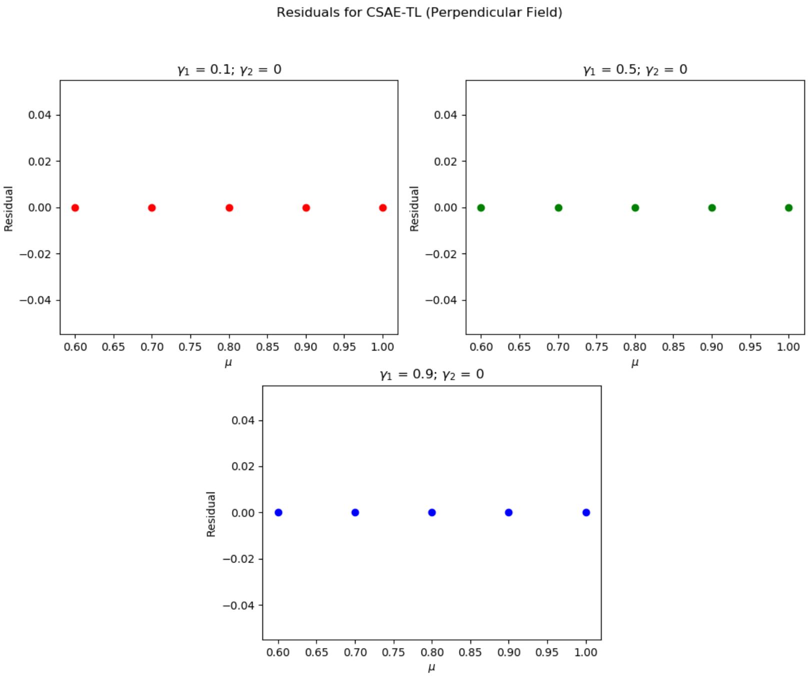

Figure 4.13: Residuals for CSAE-TL (⊥ Field). (Red: $\gamma_1 = 0.1$; Green: $\gamma_1 = 0.5$; Blue: $\gamma_1 = 0.9$.)

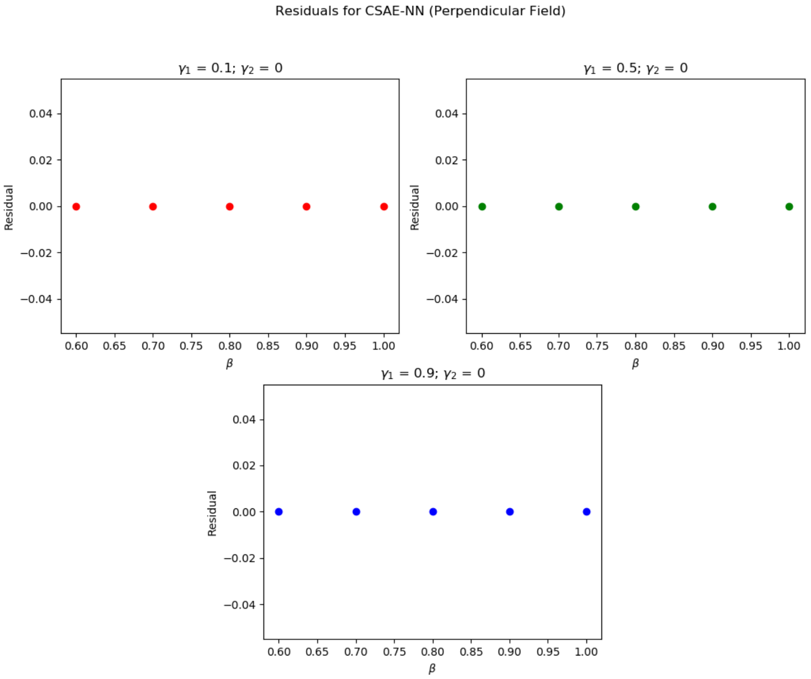

Figure 4.14: Residuals for CSAE-NN (⊥ Field). (Red: $\gamma_1 = 0.1$; Green: $\gamma_1 = 0.5$; Blue: $\gamma_1 = 0.9$.)

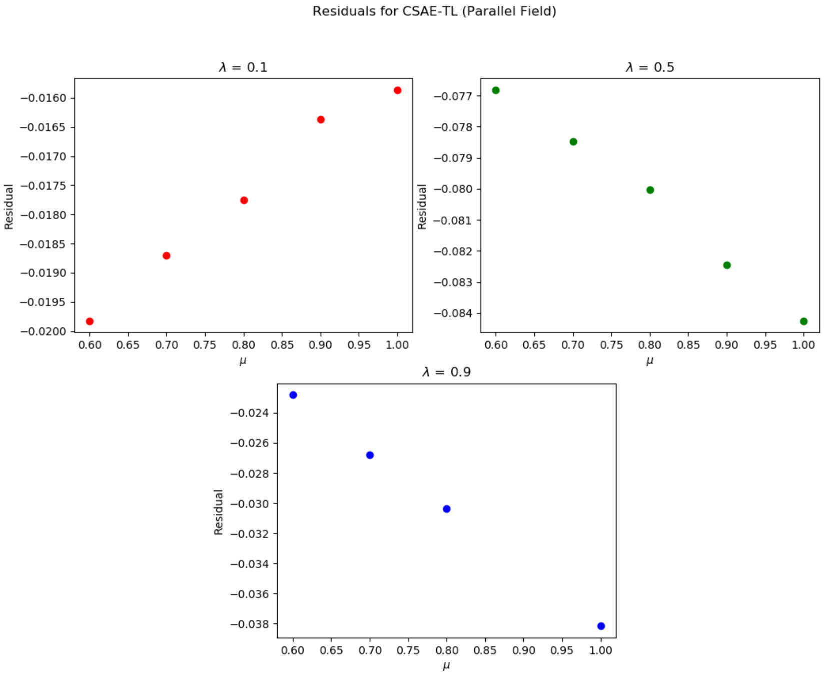

Figure 4.15: Residuals for CSAE-TL (∥ Field). (Red: $\lambda = 0.1$; Green: $\lambda = 0.5$; Blue: $\lambda = 0.9$.)

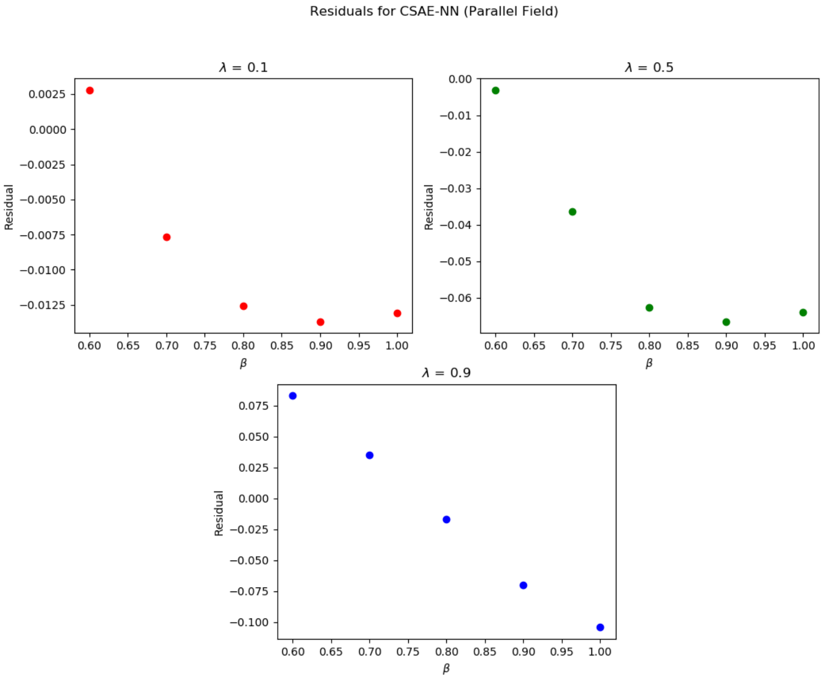

Figure 4.16: Residuals for CSAE-NN (∥ Field). (Red: $\lambda = 0.1$; Green: $\lambda = 0.5$; Blue: $\lambda = 0.9$.

Chapter 5

Ising Model Approach to ISAM

While CSAE models are particularly useful in modeling and simulating the ionic self-assembly of charged nanoparticles, there are many other mathematical frameworks capable of describing two-state systems. Another useful framework is the Ising model, originally developed by Heinrich Lenz [56] and solved by Ernst Ising [57] in an attempt to describe how phase transitions in ferromagnets emerge from a collection of individual magnetic spins. From equilibrium statistical physics, we know that the probability density of any system exchanging heat with a heat reservoir of temperature T is described by the canonical distribution:

$$P_{eq}(s) = \frac{e^{-\mathcal{H}(s)/k_B T}}{Z}, \tag{5.1}$$

where $\mathcal{H}(s)$ is the Hamiltonian of the system, k_B is Boltzmann's constant, and Z is the partition function, which normalizes the probability distribution. Lenz and Ising suggested that, for a system comprised of N interacting two-state particles in an external field $\mathbf{B}$, the Hamiltonian is given by

$$\mathcal{H} = -J \sum_{i,j \in NN} s_i s_j - B \sum_{i=1}^{N} s_i. \tag{5.2}$$

In this equation, $s_i = -1, 1$ represents the state of particle $i \in [1, N]$. In the case of ferromagnetism, the two possible states would be spin-up ($1 \sim \uparrow$) and spin-down ($-1 \sim \downarrow$); however, any two state system is possible, as we will see in our model, where s_i will come to represent the occupation state of site i on a square lattice. J is a coupling constant, which describes how two particles within the system, s_i and s_j, interact with one another.[1] The coupling constant is applied to all pairs of nearest neighbors by the first sum. Finally, B is the strength of the external field.[2] For ferromagnetism, the field is a magnetic field; however, other fields, such as electric fields, can also be used. The second sum ensures that each particle in the system feels the effects of the external field.

In time independent (i.e., equilibrium) systems, the properties of this model can be investigated by simple substitution into Equation (5.1). However, for time dependent (i.e., non-equilibrium) systems, such as ISAM, the *master equation* approach of R J Glauber [58] is necessary. Glauber begins with a master equation which ensures that the configurational properties of

[1] Generally speaking, J could be variable, indicating that different pairs of particles have different mutual interactions. This behavior is not necessary in our model; therefore, we assume J is constant.

[2] Like J, B could be variable. However, for our purposes, a constant B will suffice.

transitioning particles are conserved:

$$\frac{dP(s,t)}{dt} = \sum_{s'} \{c(s' \to s)P(s',t) - c(s \to s')P(s,t)\}. \tag{5.3}$$

Here, we see a system in which the probability of a particle existing in state s at time t depends upon the transfer of probability into state s from state s' (gain), as well as the transfer of probability from state s into state s' (loss). In spin systems, a transfer from state s to state s' would be represented by a spin flip. In ISAM, such a transfer is represented by a particle attachment site transitioning from occupied to unoccupied (or vice versa) due to evaporation or deposition.

At equilibrium, Equation (5.3) becomes

$$0 = \sum_{s'} \{c(s' \to s)P_{eq}(s',t) - c(s \to s')P_{eq}(s,t)\}. \tag{5.4}$$

Glauber's approach requires the selection of transition rates which satisfy the detailed balance condition:

$$c(s' \to s)P_{eq}(s') = c(s \to s')P_{eq}(s). \tag{5.5}$$

If we substitute the canonical distribution (Equation (5.1)) into the detailed balance condition (Equation (5.5)), we find that

$$\frac{c(s' \to s)}{c(s \to s')} = e^{\Delta H/k_B T}, \tag{5.6}$$

where $\Delta H = H(s') - H(s)$ is the change in energy which occurs any time a particle transitions from one state to the other.

We now work to translate the CSAE-NN model into a the mathematics of the Ising model, following the method outlined in [41, 59]. As a reminder to the reader, CSAE models imagine the flat surface upon which nanoparticle monolayers form as a square lattice. Each site i can be either occupied by a particle ($n_i = 1$) or unoccupied ($n_i = 0$). Empty sites can received particles (a $c(0 \to 1)$ transition) at a rate $\alpha\beta^{\eta}$, where $\eta = \sum_{j \in NN} n_j$, via the deposition process. Occupied sites can lose particles (a $c(1 \to 0)$ transition) at a rate γ. This produces the transition rate

$$c(n_i \to (1 - n_i)) = \gamma n_i + (1 - n_i)\alpha\beta^{\eta}. \tag{5.7}$$

We can express the site states $n_i = 0, 1$ from our CSAE models in terms of the spin states $s_i = -1, 1$ from the Ising model via

$$n_i = \frac{1 + s_i}{2}. \tag{5.8}$$

Our transition rate is now

$$c(s_i) = \frac{1 + s_i}{2}\gamma + \frac{1 - s_i}{2}\alpha\beta^{\sum_{j \in NN} \frac{1+s_j}{2}}. \tag{5.9}$$

If we define $K \equiv \frac{J}{k_B T}$ and $h \equiv \frac{B}{k_B T}$, the detailed balance condition becomes

$$\frac{c(s)}{c(s')} = \frac{P_{eq}(s')}{P_{eq}(s)} = \frac{e^{-K s_i \sum_{NN} s_j - h s_i}}{e^{K s_i \sum_{NN} s_j + h s_i}}. \tag{5.10}$$

s is any any of the 2^N possible configurations of the N spins contained within the system. Meanwhile, s' describes the same state with exactly one of its spins flipped. From this point, we seek K and h values appropriate to our particle assembly scenario. With these values in hand, we will know both J and B from Equation (5.2), giving us the Ising model description of ISAM.

We begin our search for K and h by supposing that each particle has $z = 2$ nearest neighbors. This describes a one-dimensional system which we will later extend into a full two-dimensional grid ($z = 4$). In this case, Equation (5.9) becomes

$$c(s_i) = \frac{1+s_i}{2}\gamma + \frac{1-s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)}. \tag{5.11}$$

Substituting into the detailed balance equation (Equation (5.10)) yields

$$\frac{\frac{1+s_i}{2}\gamma + \frac{1-s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)}}{\frac{1-s_i}{2}\gamma + \frac{1+s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)}} = \frac{e^{-Ks_i\sum_{NN}s_j - hs_i}}{e^{Ks_i\sum_{NN}s_j + hs_i}} \tag{5.12}$$

$$\frac{\frac{1+s_i}{2}\gamma + \frac{1-s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)}}{\frac{1-s_i}{2}\gamma + \frac{1+s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)}} = \frac{e^{-Ks_i(s_{i-1}+s_{i+1}) - hs_i}}{e^{Ks_i(s_{i-1}+s_{i+1}) + hs_i}}. \tag{5.13}$$

We can write this equation eight times, once for each case of the set (s_i, s_{i+1}, s_{i-1}):

$$\uparrow\uparrow\uparrow$$
$$\downarrow\uparrow\uparrow$$
$$\uparrow\downarrow\uparrow$$
$$\uparrow\uparrow\downarrow$$
$$\uparrow\downarrow\downarrow$$
$$\downarrow\downarrow\uparrow$$
$$\downarrow\uparrow\downarrow$$
$$\downarrow\downarrow\downarrow\,.$$

The resulting system of eight independent equations provides enough information to find

$$K = \frac{1}{4}\ln(\beta) \tag{5.14}$$

and

$$h = \frac{1}{2}\ln\left(\frac{\alpha\beta}{\gamma}\right). \tag{5.15}$$

Generalizing these results to a system with z nearest neighbors gives us

$$K = \frac{1}{4}\ln(\beta) \tag{5.16}$$

and

$$h = \frac{1}{4}\ln\left(\frac{\alpha^2\beta^z}{\gamma^2}\right). \tag{5.17}$$

Using $z = 4$ and the results in Equations (5.16) and (5.17), the Hamiltonian for the ISAM grid is

$$\mathcal{H} = -\frac{k_B T}{4}\left[\ln(\beta)\sum_{i,j\in NN} s_i s_j + \ln\left(\frac{\alpha^2\beta^4}{\gamma^2}\right)\sum_{i=1}^{N} s_i\right]. \tag{5.18}$$

If we define the magnetization of this system as $M \equiv \langle s_i \rangle$, as is normal in an Ising model description of a collection of spins, we finally find that the particle coverage density is given by

$$\rho = \frac{1+M}{2}. \tag{5.19}$$

For nanoparticle assembly under perpendicular electric fields, we begin with the transition rate

$$c(n_i \to (1-n_i)) = (1-n_i)\gamma_1 + n_i\gamma_2 + (1-n_i)\alpha\beta^{\eta}. \tag{5.20}$$

To find the transition rate in terms of spin value s_i, we substitute in Equation (5.8). This yields

$$c(s_i) = \frac{1-s_i}{2}\gamma_1 + \frac{1+s_i}{2}\gamma_2 + \frac{1-s_i}{2}\alpha\beta^{\sum_{j\in NN}\frac{1+s_j}{2}}. \tag{5.21}$$

When $z = 2$, this transition rate produces the following detailed balance equation:

$$\frac{\frac{1-s_i}{2}\gamma_1 + \frac{1+s_i}{2}\gamma_2 + \frac{1-s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)}}{\frac{1+s_i}{2}\gamma_1 + \frac{1-s_i}{2}\gamma_2 + \frac{1+s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)}} = \frac{e^{-Ks_i(s_{i-1}+s_{i+1})-hs_i}}{e^{Ks_i(s_{i-1}+s_{i+1})+hs_i}}. \tag{5.22}$$

The eight possible combinations of (s_{i-1}, s_i, s_{i+1}) produce a system of eight equations, which we solve to find

$$K = \frac{1}{4}\ln\left(\frac{\gamma_1 + \alpha\beta^2}{\gamma_1 + \alpha\beta}\right) \tag{5.23}$$

and

$$h = \frac{1}{2}\ln\left(\frac{\gamma_1 + \alpha\beta}{\gamma_2}\right). \tag{5.24}$$

Generalized for any z, K and h become

$$K = \frac{1}{2z}\ln\left(\frac{\gamma_1 + \alpha\beta^z}{\gamma_1 + \alpha\beta^{z/2}}\right) \tag{5.25}$$

and

$$h = \frac{1}{2}\ln\left(\frac{\gamma_1 + \alpha\beta^{z/2}}{\gamma_2}\right). \tag{5.26}$$

Notice that when $\gamma_1 = 0$ and $\gamma_2 = \gamma$ (the no-field case) Equations (5.23) - (5.26) reduce to our previous results (Equations (5.14) - (5.17)), as expected. Using $z = 4$ and the results of Equations (5.25) and (5.26), the Hamiltonian for DSAM under perpendicular electric fields is

$$\mathcal{H} = -\frac{k_B T}{2}\left[\frac{1}{4}\ln\left(\frac{\gamma_1 + \alpha\beta^4}{\gamma_1 + \alpha\beta^2}\right)\sum_{i,j\in NN} s_i s_j + \ln\left(\frac{\gamma_1 + \alpha\beta^2}{\gamma_2}\right)\sum_{i=1}^{N} s_i\right]. \tag{5.27}$$

For nanoparticle assembly under parallel electric fields, we begin with the transition rate

$$c(s_i) = n_i\gamma + (1-n_i)\alpha\beta^{\eta} + n_i(1-n_{i\pm 1})\lambda. \tag{5.28}$$

In terms of spin states, this transition rate becomes

$$c(s_i) = \frac{1+s_i}{2}\gamma + \frac{1-s_i}{2}\alpha\beta^{\sum_{j\in NN}\frac{1+s_j}{2}} + \frac{(1+s_i)(1-s_{i\pm 1})}{4}\lambda. \tag{5.29}$$

For the remainder of our treatment, we will assume that the parallel field is oriented so as to encourage particle diffusion in the positive direction. Thus, $s_{i\pm 1}$ becomes s_{i+1}.[3] The transition rate is now

$$c(s_i) = \frac{1+s_i}{2}\gamma + \frac{1-s_i}{2}\alpha\beta^{\sum_{j\in NN}\frac{1+s_j}{2}} + \frac{(1+s_i)(1-s_{i+1})}{4}\lambda. \tag{5.30}$$

When $z = 2$, this transition rate produces the following detailed balance equation:

$$\frac{\frac{1+s_i}{2}\gamma + \frac{1-s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)} + \frac{(1+s_i)(1-s_{i+1})}{4}\lambda}{\frac{1-s_i}{2}\gamma + \frac{1+s_i}{2}\alpha\beta^{\left(1+\frac{s_{i-1}+s_{i+1}}{2}\right)} + \frac{(1-s_i)(1-s_{i+1})}{4}\lambda} = \frac{e^{-Ks_i(s_{i-1}+s_{i+1})-hs_i}}{e^{Ks_i(s_{i-1}+s_{i+1})+hs_i}}. \tag{5.31}$$

Using the eight possible combinations of $(s_{i-1},\ s_i,\ s_{i+1})$, we produce a system of eight equations with solutions

$$K = \frac{1}{4}\ln(\beta) \tag{5.32}$$

and

$$h = \frac{1}{2}\ln\left(\frac{\alpha\beta}{\gamma+\lambda}\right). \tag{5.33}$$

For a generalized z, K and h become

$$K = \frac{1}{4}\ln(\beta) \tag{5.34}$$

and

$$h = \frac{1}{2}\ln\left(\frac{\alpha\beta^{z/2}}{\gamma+\lambda}\right). \tag{5.35}$$

For $\lambda = 0$ (the no-field case), Equations (5.34) and (5.35) become Equations (5.16) and (5.17), as expected. Using $z = 4$ and the results of Equations (5.34) and (5.35), the Hamiltonian for DSAM under a parallel electric field is

$$\mathcal{H} = -\frac{k_B T}{2}\left[\frac{1}{2}\ln(\beta)\sum_{i,j\in NN} s_i s_j + \ln\left(\frac{\alpha\beta^2}{\gamma+\lambda}\right)\sum_{i=1}^{N} s_i\right]. \tag{5.36}$$

[3] If the field were oriented so as to encourage particle diffusion in the negative direction, $s_{i\pm 1}$ would become s_{i-1}.

Chapter 6

Conclusion

In this thesis, we presented modeling, simulation, and experimental techniques for the production of thin films of SiO_2 nanoparticles. We examined cases in which particles assembled under no external field (ionic self-assembly of monolayers, or ISAM). We also explored thin films formed under the influence of perpendicular and parallel DC and AC fields (directed self-assembly of monolayers, or DSAM).

Our modeling approach focused on two types of cooperative sequential adsorption with evaporation (CSAE) models, which imagine the deposition surface as a discrete grid with occupied and unoccupied sites. The total lattice model (CSAE-TL) limited particle deposition according to the total number of particles deposited. The nearest neighbors model (CSAE-NN) limited particle deposition according to the total number of nearest neighbors interacting with a particular site. Both models were converted to a differential equation predicting the time dependency of the deposition surface's particle coverage density. Numerical solutions to these differential equations predicted that assembling systems would rapidly approach steady state.

Our simulation approach used the Monte Carlo method to alter the state of each site on an $(m \times n)$ grid. At each site, the program would select a random number and compare it to the result of the applicable transition rate (CSAE-TL or CSAE-NN). The results of this comparison would determine whether the site's state should change or remain the same. Like our models, simulations of ISAM and DSAM predicted a rapid approach to steady state. Residual analysis demonstrated strong quantitative agreement between our CSAE-TL models for the no field case and perpendicular field case and our simulations. Our CSAE-TL model for the parallel field case and our CSAE-NN models failed to agree quantitatively with our simulations. However, they agreed qualitatively, indicating that CSAE modeling via the mean field approximation is useful when high precision is not required or when simulation tools are unavailable. Future work could include using more accurate versions of mean field theory, including the Bethe approximation, to improve agreement between our models and simulations. Approaches of this nature generally require use of the Ising model, which we have also described for the reader in the context of thin film production.

Our experiments show that, under no electric field, the particle coverage density of a thin film can be controlled via the concentration of the nanoparticle suspension used during the ISAM process. When scaled properly, our theoretical models match experimental results for this investigation. Additionally, our investigation of the time dependence of particle coverage density shows that the assembling system does, in fact, rapidly approach steady state, just as predicted by our models and simulations. Furthermore, the increase in particle coverage density during that rapid approach can be divided into two stages, a Langmuir-type adsorption and a smooth transition to steady state. Experimental investigations of the influence of external fields on nanoparticle assembly have progressed through the experiment design stage and are ongoing. We anticipate, in light of the predictions of our models and simulations, that electric fields can be used to control particle coverage density. Future experimental work will include efforts to confirm this prediction.

Bibliography

[1] Feynman R P 1960 *Engineering and Science* **23** 22-36

[2] Lindsay S M 2010 *Introduction to Nanoscience* (Oxford: Oxford University Press)

[3] Vieu C, Carcenac F, Pépin A, Chen Y, Mejias M, Lebib A, Manin-Ferlazzo L, Couraud L, Launonis H 2000 *Appl. Surf. Sci.* **164** 111-117

[4] Tseng A A, Chen K, Chen C D, Ma K J 2003 *IEEE Trans. Electron. Packag. Manuf.* **26** 141-149

[5] Broers A N, Hoole A C F, Ryan J M 1996 *Microelectron. Eng.* **32** 131-142

[6] Lee R E 1979 *J. Vac. Sci. Technol.* **16** 164

[7] Gloersen P G 1975 *J. Vac. Sci. Technol.* **12** 2835

[8] Kumar A, Whitesides G M 1993 *Appl. Phys. Lett.* **63** 2002-2004

[9] Loo Y, Willett R L, Baldwin K W, Rogers J A 2002 *J. Am. Chem. Soc.* **124** 7654-7655

[10] Huang Y, Wu J, Yang S 2011 *Microelectron. Eng.* **88** 849-854

[11] Kilby J S 1976 *IEEE Trans. Electron Devices* **23** 648 - 654

[12] Taylor J R, Zafiratos C D, Dubson M A 2015 *Modern Physics for Scientists and Engineers* (Mill Valley, CA: University Science Books)

[13] Bennig G, Gerber C, Stoll E, Albrecht T R, Quate C F 1987 *Europhys. Lett.* **3** 1281

[14] Ohnesorge F, Binnig G 1993 *Science* **260** 1451-1456

[15] Gross L, Mohn F, Moll N, Liljeroth P, Meyer G 2009 *Science* **325** 1110-1114

[16] Custance O, Perez R, Morita S 2009 *Nat. Nanotechnol.* **4** 803-810

[17] Montemagno C, Bachand G, Stelick S, Bachand M 1999 *Nanotechnology* **10** 225-231

[18] Arthur J R 2002 *Surf. Sci.* **500** 189-217

[19] Nall J R, Lathrop J W 1957 *1957 International Electron Devices Meeting* 117

[20] Clark N A, Douglas K, Rothschild K R 1988 *Self-Assembled Nanometer Lithographic Masks and Templates and Method for Parallel Fabrication of Nanometer Scale Multi-Device Structures* 4728591

[21] Volkert C A, Minor A M 2007 *MRS Bull.* **32** 389-399

[22] Mercuri F, Maldoni M, Sgamellotti A 2012 *Nanoscale* **2** 369-379

[23] Cha J N, Birkedal H, Euliss L E, Barti M H, Wong M S Deming T J, Stucky G D 2003 *J. Am. Chem. Soc.* **125** 8285-8289

[24] Letchford K, Burt H 2007 *Eur. J. Pharm. Biopharm.* **65** 259-269

[25] Iler R K 1966 *Journal of Colloid and Interface Science* **21** 569-594

[26] Langmuir I 1941 *Method of Substance Detection* 2232539

[27] Lvov Y, Ariga K, Onda M, Ichinose I, Kunitake T 1997 *Langmuir* **13** 6195-6203

[28] Luo T M, MacDonald J C, Palamore G T R 2004 *Chem. Mater.* **16** 4916-4927

[29] Steinhart M, Goring P, Dernaika H, Prabhukaran M, Gosele U, Hempel E, Thurn-Albrecht T 2006 *Phys. Rev. Lett.* **97** 027801

[30] Yin X, Wu J, Li P, Shi M, Yang H 2016 *Chem. Nano. Mat.* **2** 37-41

[31] Xu S, Dadlani A L, Acharya S, Schindler P, Prinz F 2016 *Appl. Surf. Sci.* **367** 500-506

[32] Gorbachev I A, Goryacheva I Y, Glukhovskoy E G 2016 *J. Bionanosci.* **6** 153-156

[33] Achermann M, Petruska M A, Crooker S A, Klimov V I 2003 *J. Phys. Chem. B* **107** 13782-13787

[34] Seeman N C 1998 *Annu. Rev. Biophys. Biomol. Struct.* **27** 225-248

[35] Seeman N C 2007 *Mol. Biotechnol.* **37** 246-257

[36] Yancey S E, Zhong W, Heflin J R and Ritter A L 2006 *J. Appl. Phys.* **99** 034313

[37] Ibn-Elhaj M and Schadt M 2001 *Nature* **410** 796-799

[38] Xi J Q, Schubert M F, Kim J K, Schubert E F, Chen M, Lin A, Liu W and Smart J A 2007 *Nat. Photon.* **1** 1769

[39] Kotov N A, Dekany I, Fendler J H 1995 *J. Phys. Chem.* **99** 13065

[40] Hecht E 2017 *Optics, Fifth Edition* (New York: Pearson Education, Inc.)

[41] Mazilu D A, Mazilu I and Williams H T 2018 *From Complex to Simple - Interdisciplinary Stochastic Models* (San Rafael, CA: Morgan & Claypool Publishers)

[42] Silva N P, Menacho F P and Chorilli M 2012 *J. of Pharm.* **2** 23-30

[43] Mallouk T E and Kovtyukhova N I 2002 *Chem. Eur. J.* **8** 4355-4363

[44] Zheng Y, Lalander C H, Thai T, Dhuey S, Cabrini S and Bach U 2011 *Agnew. Chem. Int. Ed.* **50** 4398-4402

[45] Ruska E 1987 *Rev. Mod. Phys.* **59** 627-638

[46] Withers M O, Baker E, Mazilu D A, Mazilu I 2019 *J. Phys. Conf. Ser.* **1391** 012006

[47] Behrens S H, Grier D G 2001 *J. Chem. Phys.* **115** 6716-6721

[48] Cook L J, Mazilu D A, Mazilu I, Simpson B M, Schwen E M, Kim V O, Seredinski A M 2014 *Phys. Rev. E* **89** 062411

[49] Wolden C, Collins R 2015 *Photolithography Procedure* [online] Available at: `http://inside.mines.edu/impl/photo.html` [Accessed 8 May 2020]

[50] Mazilu I, Mazilu D A, Melkerson R E, Hall-Mejia E, Beck G J, Nshimyumukiza S, da Fonseca C M 2016 *Phys. Rev. E.* **93** 032803

[51] Griffiths D J 1999 *Introduction to Electrodynamics, Third Edition* (Upper Saddle River, NJ: Prentice Hall)

[52] Reese R L 2000 *University Physics* (Pacific Grove, CA: Brooks/Cole Publishing Company)

[53] Baker E, Withers M O, Aldrich E, Shaffrey I, Pusztay J, Mazilu D A, Mazilu I 2019 *J. Phys. Conf. Ser.* **1391** 012007

[54] Croarkin C, Tobias P 2012 *NIST/SEMATECH e-Handbook of Statistical Methods* (Washington, DC: National Institute of Standards and Technology) [online] Available at: `https://www.itl.nist.gov/div898/handbook/index.htm` [Accessed 6 May 2020]

[55] Gould H, Tobochnik J 2010 *Statistical and Thermal Physics: With Computer Applications* (Princeton, NJ: Princeton University Press)

[56] Lenz W 1920 *Z. Phys.* **21** 613-615

[57] Ising E 1925 *Z. Phys.* **31** 253-258

[58] Glauber R J 1963 *J. Math. Phys.* **4** 294-307

[59] Schwen E M, Mazilu I, Mazilu D A 2015 *Eur. J. Phys.* **36** 025003

[60] Schollwock U n.d. *Advanced Statistical Physics Lecture Notes* (Munich: Ludwig-Maximilians-Universitat Munchen) [online] Available at: `https://www.physik.uni-muenchen.de/lehre/vorlesungen/sose_14/asp/lectures/ASP_Chapter5.pdf` [Accessed 6 May 2020]

[61] Kinder J M, Nelson P 2018 *A Student's Guide to Python for Physical Modeling, Updated Edition* (Princeton, NJ: Princeton University Press)

[62] Hunter J, Dale D, Firing E, Droettboom M 2020 *Matplotlib.Animation.Artistanimation - Matplotlib 3.2.1 Documentation* [online] Available at: `https://matplotlib.org/api/_as_gen/matplotlib.animation.ArtistAnimation.html` [Accessed 6 May 2020]

[63] Hogg D W 2020 *Python matplotlib.animation.ArtistAnimation() Examples - Example 4* [online] Available at: `https://www.programcreek.com/python/example/96642/matplotlib.animation.ArtistAnimation` [Accessed 6 May 2020]

Appendix A

Mean Field Theory

The development of our CSAE models depends heavily upon the use of mean field theory. Throughout Chapters 2 and 3, we use mean field theory's assertion of a lack of correlation between deposition sites. This allows us to approximate the average correlation between neighboring particles as the product of the mean individual site occupations:

$$\langle n_i n_j \rangle = \langle n_i \rangle \langle n_j \rangle. \tag{A.1}$$

Ultimately, this approximation makes it possible to express our original differential equations in terms of the particle coverage density ρ. Here we consider mean field theory more generally and in its original context: the Ising model.

As discussed in [60], mean field theory depends upon the Bogolyubov inequality, which is derived as follows. First, we assume we have a system with a classical Hamiltonian $\mathcal{H}$. We now decompose this Hamiltonian into two parts:

$$\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_1. \tag{A.2}$$

While the Hamiltonian can be broken into any two parts, the approximation requires that we include all the physical information that can be solved exactly in $\mathcal{H}_0$. Similarly, we should include the more difficult parts of the Hamiltonian in $\mathcal{H}_1$. The partition function of $\mathcal{H}$ is given by

$$Z = \sum_{\{s\}} e^{-\beta \mathcal{H}(s)} \tag{A.3}$$

while the partition function of $\mathcal{H}_0$ is given by

$$Z_0 = \sum_{\{s\}} e^{-\beta \mathcal{H}_0(s)}. \tag{A.4}$$

If we decompose $\mathcal{H}$ correctly, we should be able to find Z_0 analytically. Dividing Equation (A.3) by (A.4) yields

$$\frac{Z}{Z_0} = \frac{\sum_{\{s\}} e^{-\beta(\mathcal{H}_0 + \mathcal{H}_1)}}{\sum_{\{s\}} e^{-\beta \mathcal{H}_0}} = \sum_{\{s\}} p_0(s) e^{-\beta \mathcal{H}_1(s)} = \langle e^{-\beta \mathcal{H}_1} \rangle_0. \tag{A.5}$$

Thus, by expressing the ratio of Z and Z_0 in terms of the probability distribution $p_0(s)$ associated with $\mathcal{H}_0$, we find that Z/Z_0 is the expectation value of $e^{-\beta\mathcal{H}_1}$ with respect to $p_0(s)$. The convexity inequality states that any function $f(x)$ with $f' > 0$ and $f'' > 0$ obeys

$$f(\langle x\rangle_p) \leq \langle f(x)\rangle_p. \tag{A.6}$$

p refers to the probability density $p(x)$ that we use to calculate the expectation value. Applying the convexity inequality gives us

$$\begin{aligned} \langle e^{-\beta\mathcal{H}_1}\rangle_0 &\geq e^{-\beta\langle\mathcal{H}_1\rangle_0} && \text{(A.7)}\\ \frac{Z}{Z_0} &\geq e^{-\beta\langle\mathcal{H}_1\rangle_0}. && \text{(A.8)} \end{aligned}$$

Taking the logarithm of this function and using $G = -k_BT\text{ln}Z$, the definition of Gibbs free energy, finally gives us the Bogolyubov inequality:

$$G \leq G_0 + \langle\mathcal{H}_1\rangle_0. \tag{A.9}$$

When splitting our Hamiltonian, we use a control parameter λ. The mean field approximation arises when we minimize the Gibbs free energy with respect to this parameter. We begin with

$$\mathcal{H}_0^{(\lambda)} = -\lambda\sum_i s_i \tag{A.10}$$

and

$$\mathcal{H}_1^{(\lambda)} = -J\sum_{\{i,j\}} s_is_j + (\lambda - H)\sum_i s_i. \tag{A.11}$$

Since we consider a two state ising model, $s_i = \pm 1$. This means that the partition function for $\mathcal{H}_0^{(\lambda)}$ becomes

$$\begin{aligned} Z_0 &= \left(e^{-\beta\lambda} + e^{\beta\lambda}\right)^N && \text{(A.12)}\\ &= (2\text{cosh}(\beta\lambda))^N && \text{(A.13)} \end{aligned}$$

where N is the number of particles in the system. The Gibbs free energy associated with $\mathcal{H}_0^{(\lambda)}$ is

$$\begin{aligned} G_0 &= -k_BT\text{ln}(Z_0) && \text{(A.14)}\\ &= -Nk_BT\text{ln}(2\text{cosh}(\beta\lambda)). && \text{(A.15)} \end{aligned}$$

The expectation values of the spins with respect to $\mathcal{H}_0^{(\lambda)}$ are independent of one another. This means that

$$\langle s_i\rangle_0 = \tanh(\beta\lambda) \tag{A.16}$$

for all values of index i. Independence also produces the following result for the expectation value of $\mathcal{H}_1^{(\lambda)}$:

$$\left\langle\mathcal{H}_1^{(\lambda)}\right\rangle = -\frac{1}{2}NJz\text{tanh}^2(\beta\lambda) + N(\lambda - H)\text{tanh}(\beta\lambda), \tag{A.17}$$

where z is the number of nearest neighbors. Plugging Equations (A.15) and (A.17) into Equation (A.9) tells us that we should minimize

$$G(\lambda) = N\Big[- \beta^{-1}\ln(2\cosh(\beta\lambda)) - \frac{1}{2}Jz\tanh^2(\beta\lambda) + (\lambda - H)\tanh(\beta\lambda)\Big]. \tag{A.18}$$

Computing $\mathrm{d}G/\mathrm{d}\lambda = 0$ produces

$$\lambda_{\mathrm{min}} - H = Jz\tanh(\beta\lambda_{\mathrm{min}}), \tag{A.19}$$

where λ_{min} is the value of λ at which the Gibbs free energy is minimized. The minimized free energy is

$$G = -Nk_bT\ln(2\cosh(\beta\lambda_{\mathrm{min}})) + \frac{N(\lambda_{\mathrm{min}} - H)^2}{2zJ}. \tag{A.20}$$

The magnetization of the system is given by

$$\begin{align} m &= -\frac{1}{N}\frac{\mathrm{d}G}{\mathrm{d}H} \tag{A.21} \\ &= -\frac{1}{N}\left(\frac{\partial G}{\partial H} + \frac{\partial G}{\partial \lambda}\frac{\partial \lambda}{\partial H}\right)_{\lambda_{\mathrm{min}}} \tag{A.22} \end{align}$$

Since the free energy is minimized, $\frac{\partial G}{\partial \lambda} = 0$ and $\lambda = \lambda_{\mathrm{min}}$. We now have

$$m = \frac{\lambda_{\mathrm{min}} - H}{Jz}. \tag{A.23}$$

Substituting in Equation (A.19) finally yields

$$m = \tanh\big[\beta(Jzm + H)\big]. \tag{A.24}$$

According to [55], this is an example of a self-consistent equation for m. In other words, the mean field that produces $\langle m\rangle$ depends upon $\langle m\rangle$.

When H = 0, which represents a system in which there is no external magnetic field[1], Equation (A.24) becomes

$$m = \tanh(\beta Jzm). \tag{A.25}$$

Stable solutions to this transcendental equation exist only when the system is above a critical temperature T_c given by

$$T_c = \frac{Jz}{k_B}. \tag{A.26}$$

When T is close to T_c, m is necessarily small. This allows us to Taylor expand ($\tanh x \approx x - x^3/3 + \cdots$ when $x << 1$) Equation (A.25) to find

$$m = \beta Jzm - \frac{1}{3}(\beta Jzm)^3 + \cdots . \tag{A.27}$$

[1] Each spin still experiences the mean magnetic field produced by all the other spins.

The solutions are

$$m(T > T_c) = 0 \tag{A.28}$$

and

$$m(T < T_c) = \pm \frac{3^{1/2}}{(\beta J z)^{3/2}} (\beta J z - 1)^{1/2}. \tag{A.29}$$

Mean field theory's predictions for critical temperature (T_{mf}) often vary from exact values / best known-estimates (T_c). Table A.1 demonstrates this idea for lattices with differing numbers of spatial dimensions (d) and nearest neighbors (z).

Lattice Type	d	z	T_{mf}/T_c
Square	2	4	1.763
Hexagonal	2	6	1.684
Diamond	3	4	1.479
Simple Cubic	3	6	1.330
Body-Centered Cubic	3	8	1.260
Face-Centered Cubic	3	12	1.225

Table A.1: Comparison of critical temperature values produced by mean field theory T_{mf} and best-known estimates T_c for several different lattice types. (Reproduced from [55].)

The often large discrepancy between T_{mf} and the actual T_c arises for two main reasons. First, mean field theory depends only on the number of nearest neighbors z and not on the dimensionality d of the system. Both parameters must be included in an accurate treatment of spin systems. Second, mean field theory ignores correlations between spins. However, mean field theory still produces results that are qualitatively correct. For this reason, it is an important tool in the analysis of many-particle, multi-state systems, especially those that do not have exact solutions.

Appendix B

Python Tools

B.1 odeint

odeint is a function from scipy.integrate which allows users to find numerical solutions to first-order ordinary differential equations (ODEs) [61]. The function uses the initial configuration of a system to iteratively calculate succeeding configurations. This process outputs a graph of the dependent variable versus the independent variable. For most physical systems, the independent variable is time, allowing the user to determine the state of a system described by a first-order ODE at any time. While odeint can only process first-order equations, it can accept systems of ODEs. Thus, we can use Python to consider higher-order equations by simply expressing them as systems of first-order equations.

odeint is imported using

```
from scipy.integrate import odeint
```

and called using

```
y = odeint(F, y0, t)
```

F is a Python function F(y, t) which accepts a one-dimensional array (y) that describes the current state of the system and a scalar (t) which describes the current value of the independent variable. y0 is a one-dimensional array that describes the initial state of the system (initial value of y). t is the array of independent variable values, the first of which corresponds to y0. Each subsequent value in t corresponds to subsequent states of the system, which are successively stored in y. Below we include a sample program from [61], which is designed to find a numerical solution of the position of a simple harmonic oscillator:

```
#solve_ode.py
"""ODE solver for harmonic oscillator."""

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# Define Oscillator Function
def F(y, t):
    ”””
    Return derivatives of second-order ODE y'' = -y.
    ”””

    dy = [0, 0]          #Create a list to store derivatives.
    dy[0] = y[1]                #Store first derivative of y(t).
    dy[1] = -y[0]               #Store second derivative of y(t).

    return dy

#Create array of time values to study
t_min = 0
t_max = 10
dt = 0.1
t= np.arange(t_min, t_max, dt)

#Set initial conditions
y0 = (1.0, 0.0)

plt.figure()                    #Create figure; add plots later.
y = odeint(F, y0, t)            #Call odeint. Pass it the function, the
    initial conditions, and the time array.
plt.plot(t, y[:, 0], linewidth=2)       #Plot the resulting numerical
    solution.
plt.title(”Numerical Solution of SHO”)
plt.xlabel(”t”)
plt.ylabel(”y”)
```

The sample program passes the function describing the harmonic oscillator (a system of first-order ODEs), the initial conditions, and the time array to `odeint`. `odeint` then returns an array, stored in y, which contains the numerical solutions to both equations in the system. The program concludes by plotting the solution to the first equation against time, as Figure B.1 demonstrates.

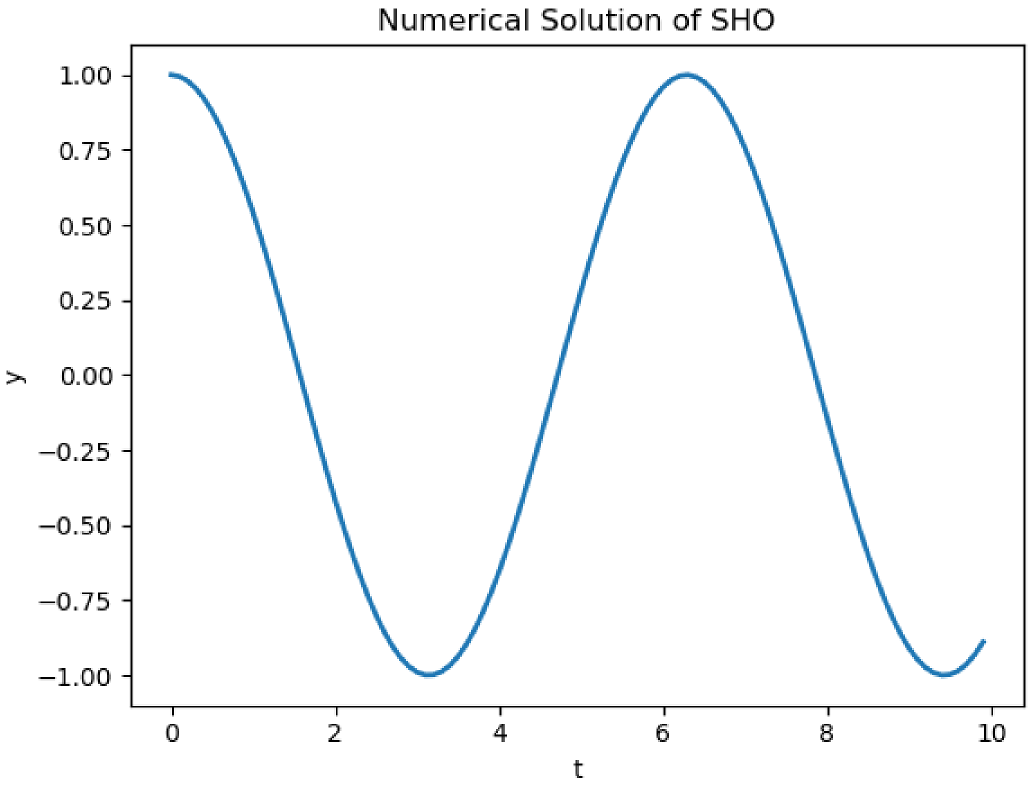

Figure B.1: Numerical solution for the position of a simple harmonic oscillator produced by sample program *solve_ode.py*.

B.2 `ArtistAnimation`

`ArtistAnimation` is a function from `matplotlib.animation` which allows users to combine a series of images or figures into an animated video [62]. Each image becomes a frame. These frames, when cycled successively, form a video file. `ArtistAnimation` is imported using

```
import matplotlib.animation as animation
```

and called using

```
im_ani = animation.ArtistAnimation(fig, ims, interval=100,
    repeat_delay=3000, blit=True)
```

fig is the figure object used to display each image. ims is the array holding the information needed to produce each image. interval=100 provides the delay between frames in milliseconds. If not included, this parameter defaults to 200. repeat_delay=3000 specifies the number of milliseconds the video pauses before repeating. If not specified, this parameter defaults to `None`. Finally, blit=True turns on blitting, an image optimization tool. If not specified, it takes on a value of `False`. Below we include a sample function from [63] which uses `ArtistAnimation` to create a video from images stored in a directory:

```
def create_video_from_model_images():
    ”””
    function loads the images of the model and concatenates them to a
      video file
    ”””
    images_path = glob.glob(”./model_*.png”)
    print(”number of image files to process: %s” % len(images_path))

    fig = plt.figure()
    ims = []
    for i in range(len(images_path)):
        print(i)
        img=mpimg.imread(images_path[i])
        lum_img = img[:,:,0]
        ims.append((plt.pcolormesh(lum_img[60:500, 180:640], cmap=’gray
          ’),))
    print(”loaded all images”)

    im_ani = animation.ArtistAnimation(fig, ims, interval=100,
                                       repeat_delay=3000, blit=True)
    print(”created the video”)
    im_ani.save(’./model_vid.mp4’)
    print(”done, the video is ready!”)
```

The sample program collects a series of images into an array named ims. It then uses the `ArtistAnimation` function to create a video which shows these stored images in rapid succession, producing an animation effect.

Appendix C

Code Samples

C.1 CSAE-TL Numerical Solutions

```
# Author: Matthew O. Withers

import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Defines the CSAE-TL model mean field equation with electric field
def model_diffusion(rho,t):
    gamma = 0.5
    mu = 0.5
    lamb = 0.7

    drhodt = -gamma*rho + mu*(1-rho)**2 + lamb*(rho-rho**2)
    return drhodt

#Defines the CSAE-TL model mean field equation without electric field
def model_no_diffusion(rho,t):
    gamma = 0.5
    mu = 0.5
    drhodt = -gamma*rho + mu*(1-rho)**2
    return drhodt

# Defines the initial particle density
rho0 = 0

# Defines the solution time range
t = np.linspace(0,100, num = 200)

# Solves both mean field equations
rho_diffusion = odeint(model_diffusion,rho0,t)
```

```
rho_no_diffusion = odeint(model_no_diffusion,rho0,t)

# Plots both solutions
plt.plot(t,rho_diffusion*100,label=‘Diffusion (lamb = 0.7)’)
plt.plot(t,rho_no_diffusion*100,label=‘No Diffusion (lamb = 0.0)’)
plt.xlabel(‘Time Steps (t)’)
plt.ylabel(‘Particle Coverage Density (rho(t)) [%]’)
plt.legend()
plt.title(‘Particle Coverage Density over Time (TL Model)’)
plt.show()

# Exports (Time, Coverage Density) for the electric field equation to a
    CSV file
excel_1 = np.column_stack((t,rho_diffusion))
np.savetxt(‘meanfield_tl_diffusion_output.csv’,excel_1,delimiter=‘,’)

excel_2 = np.column_stack((t,rho_no_diffusion))
np.savetxt(‘meanfield_tl_no_diffusion_output.csv’,excel_2,delimiter=‘,’
   )
```

C.2 CSAE-NN Numerical Solutions

```
# Author: Matthew O. Withers

import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Defines the CSAE-NN model mean field equation with electric field
def model_diffusion(rho,t):
    gamma = 0.5
    alpha = 0.5
    beta = 0.5
    z = 4
    lamb = 0.7

    drhodt = -gamma*rho + (1-rho)*alpha*beta**(z*rho) + lamb*(rho-rho
       **2)
    return drhodt

#Defines the CSAE-NN model mean field equation without electric field
def model_no_diffusion(rho,t):
    gamma = 0.5
    alpha = 0.5
    beta = 1
    z = 4
    drhodt = -gamma*rho + (1-rho)*alpha*beta**(z*rho)
    return drhodt

# Defines the initial particle density
rho0 = 0

# Defines the solution time range
t = np.linspace(0,100,100)

# Solves both mean field equations
rho_diffusion = odeint(model_diffusion,rho0,t)
rho_no_diffusion = odeint(model_no_diffusion,rho0,t)

# Plots both solutions
plt.plot(t,rho_diffusion*100,label=‘Diffusion (lamb = 0.7)’)
plt.plot(t,rho_no_diffusion*100,label=‘No Diffusion (lamb = 0.0)’)
plt.xlabel(‘Time Steps (t)’)
plt.ylabel(‘Particle Coverage Density (rho(t)) [%]’)
plt.legend()
plt.title(‘Particle Coverage Density over Time (NN Model)’)
plt.show()

# Exports (Time, Coverage Density) for the electric field equation to a
    CSV file
excel_1 = np.column_stack((t,rho_diffusion))
np.savetxt(‘meanfield_nn_diffusion_output.csv’,excel_1,delimiter=‘,’)

excel_2 = np.column_stack((t,rho_no_diffusion))
np.savetxt(‘meanfield_nn_no_diffusion_output.csv’,excel_2,delimiter=‘,’
    )
```

C.3 CSAE-TL Simulations

```
# Author: Matthew O. Withers

import numpy as np
import matplotlib.pyplot as plt
import random

# Defines the slide size, number of trials, and number of time steps
   per simulation
# Note: The program currently handles only square slides and 1 trial
Grid_Length = 10
Grid_Width = 10
```

```
trials = 1
timesteps = 100

# Defines model coefficients
mu = 0.5
gamma = 0.5
lamb = 0.7

# Defines storage arrays
density = np.zeros(timesteps)
time = np.zeros(timesteps)
color_map = []

# Creates the initial grid (matrix); all sites = 0
d = np.zeros((Grid_Length,Grid_Width))
d_display = np.zeros((Grid_Length,Grid_Width))

# Defines the boundaries of the slide (completely surrounded by
   particles)
for i in range(Grid_Length):
    for j in range(Grid_Width):
        d[i,0]=1
        d[i,Grid_Width-1]=1
        d[0,j]=1
        d[Grid_Length-1,j]=1

# Main simulation loop
for current_trial in range(trials):
    for current_time in range(timesteps):

        occupied = 0
        d_previous = d

        # Caculates the number of particles on the slide
        for i in range(Grid_Length):
            for j in range(Grid_Width):
                if i != 0 and i != Grid_Length - 1 and j != 0 and j !=
                   Grid_Length - 1:
                    if d_previous[i,j] == 1:
                        occupied = occupied + 1

        # Calculates and stores the coverage density and time
        density[current_time] = occupied/((Grid_Length-2)*(Grid_Width
           -2))
        time[current_time] = current_time
```

```
# Evaluates each site
for i in range(Grid_Length):
    for j in range(Grid_Width):

        # Grabs a random seed value
        simulation_number = random.uniform(0,1)

        # Limits evaluation to active slide area (no edges)
        if i != 0 and i != (Grid_Length - 1) and j != 0 and j
           != (Grid_Length - 1):

            # Determines the deposition/evaporation probability
            probability = gamma*d_previous[i,j] + mu*(1-
               d_previous[i,j])*(1-occupied/((Grid_Length-2)*(
               Grid_Width-2)))

            # Determines the diffusion probability (both
               receiving and donating)
            diffusion_donate_probability = lamb*d_previous[i,j
               ]*(1-d_previous[i,j-1])
            diffusion_receive_probability = lamb*d_previous[i,j
               +1]*(1-d_previous[i,j])

            # Compares random seed value to probabilities to
               determine new state of site
            if d_previous[i,j] == 0 and simulation_number <=
               probability:
                d[i,j] = 1
            elif d_previous[i,j] == 1 and simulation_number <=
               probability:
                d[i,j] = 0
            else:
                if d[i,j] == 0 and d[i,j+1] == 1 and
                   simulation_number <=
                   diffusion_receive_probability:
                    d[i,j] = 1
                    d[i,j+1] = 0
                elif d[i,j] == 1 and d[i,j-1] == 0 and
                   simulation_number <=
                   diffusion_donate_probability:
                    d[i,j] = 0
                    d[i,j-1] = 1
                else:
                    d[i,j] = d_previous[i,j]
        # Resets the grid boundaries
        for i in range(Grid_Length):
            for j in range(Grid_Width):
                d[i,0]=1
                d[i,Grid_Width-1]=1
                d[0,j]=1
                d[Grid_Length-1,j]=1

# Prepares the final matrix for display
for i in range(Grid_Length):
    for j in range(Grid_Width):
        if d[i,j] == 0:
            d_display[i,j] = 1
        if d[i,j] == 1:
            d_display[i,j] = 0

# Plots an image of the final slide state
plt.figure(1)
color_map = plt.imshow(d_display)
color_map.set_cmap(‘‘Blues_r”)
plt.grid()
plt.xticks(np.arange(-0.5, Grid_Width-0.5, 1))
plt.yticks(np.arange(-0.5, Grid_Length-0.5, 1))

# Plots the particle coverage density over time
plt.figure(2)
plt.plot(time,density*100)
plt.title(‘Particle Coverage Density over Time (TL Simulation)’)
plt.xlabel(‘Time Steps (t)’)
plt.ylabel(‘Particle Coverage Density (  (t)) [%]’)

# Displays plots
plt.show()

# Exports coverage density vs. time data to a CSV file
excel = np.column_stack((time,density))
np.savetxt(‘Simulation_TL_Output.csv’, excel, delimiter=‘,’)
```

C.4 CSAE-NN Simulations

```
# Author: Matthew O. Withers

import numpy as np
import matplotlib.pyplot as plt
import random

# Defines the slide size, number of trials, and number of time steps
    per simulation
# Note: The program currently handles only square slides and 1 trial
Grid_Length = 10
Grid_Width = 10
trials = 1
timesteps = 100

# Defines model coefficients
alpha = 0.5
beta = 0.5
gamma = 0.5
lamb = 0.7

# Defines storage arrays
density = np.zeros(timesteps)
time = np.zeros(timesteps)
color_map = []

# Creates the initial grid (matrix); all sites = 0
d = np.zeros((Grid_Length,Grid_Width))
d_display = np.zeros((Grid_Length,Grid_Width))

# Defines the boundaries of the slide (completely surrounded by
    particles)
for i in range(Grid_Length):
    for j in range(Grid_Width):
        d[i,0]=1
        d[i,Grid_Width-1]=1
        d[0,j]=1
        d[Grid_Length-1,j]=1

# Main simulation loop
for current_trial in range(trials):
    for current_time in range(timesteps):

        occupied = 0
        d_previous = d

        # Caculates the number of particles on the slide
        for i in range(Grid_Length):
            for j in range(Grid_Width):
                if i != 0 and i != Grid_Length - 1 and j != 0 and j !=
                   Grid_Length - 1:
                    if d_previous[i,j] == 1:
                        occupied = occupied + 1

        # Calculates and stores the coverage density and time
        density[current_time] = occupied/((Grid_Length-2)*(Grid_Width
           -2))
        time[current_time] = current_time

        # Evaluates each site
        for i in range(Grid_Length):
            for j in range(Grid_Width):

                # Grabs a random seed value
                simulation_number = random.uniform(0,1)

                # Limits evaluation to active slide area (no edges)
                if i != 0 and i != (Grid_Length - 1) and j != 0 and j
                   != (Grid_Length - 1):

                    # Determines the number of nearest neighbors for
                       the site
                    eta = 0
                    if d_previous[i+1,j] == 1:
                        eta = eta+1
                    if d_previous[i-1,j] == 1:
                        eta = eta+1
                    if d_previous[i,j+1] == 1:
                        eta = eta+1
                    if d_previous[i,j-1] == 1:
                        eta = eta+1

                    # Determines the deposition/evaporation probability
                    probability = gamma*d_previous[i,j] + (1-d_previous
                       [i,j])*alpha*beta**eta

                    # Determines the diffusion probability (both
                       receiving and donating)
                    diffusion_donate_probability = lamb*d_previous[i,j
                       ]*(1-d_previous[i,j-1])
                    diffusion_receive_probability = lamb*d_previous[i,j
                       +1]*(1-d_previous[i,j])

                    # Compares random seed value to probabilities to
                       determine new state of site
```

```
                    if d_previous[i,j] == 0 and simulation_number <=
                       probability:
                        d[i,j] = 1
                    elif d_previous[i,j] == 1 and simulation_number <=
                       probability:
                        d[i,j] = 0
                    else:
                        if d[i,j] == 0 and d[i,j+1] == 1 and
                           simulation_number <=
                           diffusion_receive_probability:
                            d[i,j] = 1
                            d[i,j+1] = 0
                        elif d[i,j] == 1 and d[i,j-1] == 0 and
                           simulation_number <=
                           diffusion_donate_probability:
                            d[i,j] = 0
                            d[i,j-1] = 1
                        else:
                            d[i,j] = d_previous[i,j]

        # Resets the grid boundaries
        for i in range(Grid_Length):
            for j in range(Grid_Width):
                d[i,0]=1
                d[i,Grid_Width-1]=1
                d[0,j]=1
                d[Grid_Length-1,j]=1

# Prepares the final matrix for display
for i in range(Grid_Length):
    for j in range(Grid_Width):
        if d[i,j] == 0:
            d_display[i,j] = 1
        if d[i,j] == 1:
            d_display[i,j] = 0

# Plots an image of the final slide state
plt.figure(1)
color_map = plt.imshow(d_display)
color_map.set_cmap(‘‘Blues_r”)
plt.grid()
plt.xticks(np.arange(-0.5, Grid_Width-0.5, 1))
plt.yticks(np.arange(-0.5, Grid_Length-0.5, 1))

# Plots the particle coverage density over time
```

```
plt.figure(2)
plt.plot(time,density)
plt.title(‘Particle Coverage Density over Time (NN Simulation)’)
plt.xlabel(‘Time Steps (t)’)
plt.ylabel(‘Particle Coverage Density (  (t)) [%]’)

# Displays plots
plt.show()

# Exports coverage density vs. time data to a CSV file
excel = np.column_stack((time,density))
np.savetxt(‘Simulation_NN_Output.csv’, excel, delimiter=‘,’)
```

C.5 Simulation Video Output

```
# Author: Matthew O. Withers

import numpy as np
import matplotlib.pyplot as plt
import matplotlib
import random
import matplotlib.animation as animation
matplotlib.use(‘‘Agg”)

# Defines the slide size, number of trials, and number of time steps
    per simulation
# Note: The program currently handles only square slides and 1 trial
Grid_Length = 10
Grid_Width = 10
trials = 1
timesteps = 50

# Defines model coefficients
mu = 0.7
gamma = 0.2
lamb = 0.8

# Defines storage arrays
density = np.zeros(timesteps)
time = np.zeros(timesteps)
color_map = []
ims = []

# Creates the initial grid (matrix); all sites = 0
d = np.zeros((Grid_Length,Grid_Width))
```

```
d_display = np.zeros((Grid_Length,Grid_Width))

# Defines the boundaries of the slide (completely surrounded by
   particles)
for i in range(Grid_Length):
    for j in range(Grid_Width):
        d[i,0]=1
        d[i,Grid_Width-1]=1
        d[0,j]=1
        d[Grid_Length-1,j]=1

# Main simulation loop
for current_trial in range(trials):
    for current_time in range(timesteps):

        occupied = 0
        d_previous = d

        # Caculates the number of particles on the slide
        for i in range(Grid_Length):
            for j in range(Grid_Width):
                if i != 0 and i != Grid_Length - 1 and j != 0 and j !=
                   Grid_Length - 1:
                    if d_previous[i,j] == 1:
                        occupied = occupied + 1

        # Calculates and stores the coverage density and time
        density[current_time] = occupied/((Grid_Length-2)*(Grid_Width
           -2))
        time[current_time] = current_time

        # Evaluates each site
        for i in range(Grid_Length):
            for j in range(Grid_Width):

                # Grabs a random seed value
                simulation_number = random.uniform(0,1)

                # Limits evaluation to active slide area (no edges)
                if i != 0 and i != (Grid_Length - 1) and j != 0 and j
                   != (Grid_Length - 1):

                    # Determines the deposition/evaporation probability
                    probability = gamma*d_previous[i,j] + mu*(1-
                       d_previous[i,j])*(1-occupied/((Grid_Length-2)*(
                       Grid_Width-2)))

                    # Determines the diffusion probability (both
                       receiving and donating)
                    diffusion_donate_probability = lamb*d_previous[i,j
                       ]*(1-d_previous[i,j-1])
                    diffusion_receive_probability = lamb*d_previous[i,j
                       +1]*(1-d_previous[i,j])

                    # Compares random seed value to probabilities to
                       determine new state of site
                    if d_previous[i,j] == 0 and simulation_number <=
                       probability:
                        d[i,j] = 1
                    elif d_previous[i,j] == 1 and simulation_number <=
                       probability:
                        d[i,j] = 0
                    else:
                        if d[i,j] == 0 and d[i,j+1] == 1 and
                           simulation_number <=
                           diffusion_receive_probability:
                            d[i,j] = 1
                            d[i,j+1] = 0
                        elif d[i,j] == 1 and d[i,j-1] == 0 and
                           simulation_number <=
                           diffusion_donate_probability:
                            d[i,j] = 0
                            d[i,j-1] = 1
                        else:
                            d[i,j] = d_previous[i,j]

        # Resets the grid boundaries
        for i in range(Grid_Length):
            for j in range(Grid_Width):
                d[i,0]=1
                d[i,Grid_Width-1]=1
                d[0,j]=1
                d[Grid_Length-1,j]=1

        # Prepares the current matrix for display
        for i in range(Grid_Length):
            for j in range(Grid_Width):
                if d[i,j] == 0:
                    d_display[i,j] = 1
                if d[i,j] == 1:
```

```
                    d_display[i,j] = 0

        # Creates a figure reference
        fig = plt.figure(1)

        # Stores the current matrix as an image in the animation array
        im = plt.imshow(d_display, animated=True)
        im.set_cmap(‘‘Blues_r”)
        plt.grid()
        plt.xticks(np.arange(0, Grid_Width, 0.5))
        plt.yticks(np.arange(0, Grid_Length, 0.5))
        ims.append([im])

# Builds a animation out of all images in the animation array
ani = animation.ArtistAnimation(fig, ims, interval=50, blit=True,
                                repeat_delay=1000)

# Saves the animation as an MP4 file
ani.save(‘dynamic_images.mp4’)

# Plots the coverage density vs. time
plt.figure(2)
plt.plot(time,density)

# Displays animation and graph
plt.show()
```