Optical Properties of Self-Assembled Nanoparticle Thin Films

K. Elise Baker

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
2021

Abstract

In this thesis, we explore the optical properties of silicon-dioxide (silica) thin-films generated through the use of a process known as *ionic self-assembly of monolayers* (ISAM). This process may be repeated for multiple bilayers with the ultimate goal of producing multilayer thin films with desirable optical properties. We model these results mathematically using cooperative sequential adsorption with evaporation (CSAE) models. These models can accurately predict the particle density of multiple layer nanoparticle structures, which can then be used to predict the index of refraction of a multiple layer structure. In addition, we explore the effect of external electric fields applied across the glass substrate during the dipping process, as well as the behavior of titania nanoparticles in single layer thin films.

Acknowledgments

Firstly, I would like to thank my thesis advisor, Dr. Dan Mazilu. Not only has he acted as a thesis advisor over the course of this year, he has also acted as my research advisor, professor, mentor, and friend. I would like to thank him for all of the time he has dedicated over the past four years, specifically this past year, to furthering my learning, and pushing me to be the best physicist I can be.

Secondly, I would like to thank my academic advisor, Dr. Irina Mazilu for all of her assistance in the course of writing this thesis, as well as her support over the past four years as my advisor, professor, mentor, and friend. Through the course of writing this thesis her support has been invaluable, particularly in writing and working on the simulation portions of this thesis.

I would also like to thank Dr. Laurentiu Stoleriu for his help in creating the Python scripts to run the simulations, and including wonderful graphics.

Additionally, I would like to thank Dave Pfaff for all of his help in the IQ Center, and for his guidance and SEM training. I would also like to thank Chris Compton for his help in the lab, and in setting up equipment for us to use in experiments.

Finally, I would like to thank all of the students I have worked in conjunction with doing research over the past three years - Matthew Withers, Cory Morris, Nolan Zunk, and Ben Zeman. I would like to extend an additional thank you to Matthew Withers for providing some of the graphics used in this thesis, as well as for being a wonderful role model in the lab.

A portion of this thesis was presented at the 8th International Conference on Mathematical Modeling in Physical Sciences in Bratislava, Slovakia. The conference was fully funded by the Avis P. Waring Scholarship, for which I am very grateful.

Contents

Chapter 1

Background and Introduction

1.1 Background

The field of nanoscience, which is the study of systems acting on the nanometer scale, was first brought to light by Richard Feynman, a renowned physicist known for his work on quantum physics [1]. During the 1959 American Physical Society meeting he gave a talk entitled "There is Plenty of Room at the Bottom." During this talk, he discussed the vast set of implications of manipulating atoms on the nanoscale. Such implications are for data storage, as well as the rise of electronics, particularly nanoscale circuitry. Feynman goes on to indicate that manipulating and creating things at the nanoscale is very feasible, as a Hydrogen atom is roughly 1/10th of a nanometer, showcasing the possibility for creating very small objects.

Feynman posed the question, "why cannot we write the entire 24 volumes of the Encyclopedia Britannica on the head of a pin?" Miniaturization had previously showed that the Lord's Prayer could be written on the head of a pin, so what is dictating that the same cannot be done for the Encyclopedia Britannica? Feynman calculated that in order to do this, the words would have to be 25,000 times smaller than they are already written to fit on the 16th inch diameter pin head. At this size, the words are roughly 80 Angstroms, or 32 atoms, across, so it is entirely feasible to do so.

Now that it has been shown that there is sufficient room to do it, how do we actually do it? Feynman proposed that we reverse the process of an electron microscope, and instead of magnifying, make it send ions in reverse to create raised metal bumps that spell out the entire Encyclopedia. In order to now read the miniaturized version, one could make a mold of the pin head, and evaporate silica into this mold, creating a thin film with the words engraved in it. Using a gold coating on the mold, the words can then be read using an electron microscope.

This kind of technology, if developed, would have the ability to save an immense amount of space. Feynman went on to propose that we use this technique to write every book in the world at this scale. He estimated that there are around 24 million volumes in the world based on the number of volumes in the most notable libraries around the world at that time. At this scale, all 24 million volumes could be written in the space that would take up around 35 pages of the original Encyclopedia Britannica, a profound discovery. Little does Feynman know that we could access every volume in the world from a device that fits in the palm of our hands today.

Not only does nanoscale work have immense possibilities for increasing storage, but it also has remarkable implications for advances in electronics as well as the medical field. At the time of

Feynman's talk, computers took up entire rooms, but Feynman proposed that by making individual wires mere atoms in diameter, and by substantially reducing the space between wires on a circuit board, this could considerably reduce the space taken to create such machines. As this thesis is written on a computer roughly the size of a pad of paper, it is clear that some of his ideas were able to come to fruition with significant advances in technology over the past 72 years.

In terms of medical advances Feynman had considered, he proposed the concept of miniaturizing a surgeon to a size that a patient could swallow it and perform the surgery from inside of the patient. While this in itself is not possible now, or likely ever, there have been many advances in the biomedical engineering field which have made minimally invasive surgery possible. There now exist endoscopic options for surgery which allow surgeons to non-invasively perform surgery on a patient, in which a device is inserted through a small incision on the patient. By manipulating a computer or other device outside of the patient, the surgeon is able to perform the surgery. Without significant advances in nanoscience, none of the concepts Feynman proposed would have been feasible.

In his textbook on nanoscience, Lindsay discusses the ways in which some of the above methods could be implemented [1]. There are two primary methods for generating nanostructures – a top-down method, and a bottom-up method. The top-down method of creating nanostructures relies on etching away material until the desired nanostructure has been created. This is done in the same way that a sculptor would create a statue. Through the use of lithography, (generally photolithography or electron beam lithography), material is able to be etched away to generate complex nanostructures. This method of production is the method by which most nanotechnology has been created. The other method for creating nanostructures is via the bottom-up approach, which is the approach used in this thesis. This method relies on the self-assembly of chemical components into the desired nanostructures.

The applications of nanoscience and nanoscale work are far reaching, ranging from anti-reflective (AR) coatings [2] [3], to drug delivery [6], nanoprinting [8] and nanocircuitry [7]. This talk by Feynman was not very well recognized at the time, and is now a key talk for people researching nanoscience. The vast implications discussed by Feynman and Lindsay make studying nanoscience incredibly important.

1.2 Molecular Monolayers and Bilayers

In this thesis we look at the optical properties of SiO_2 (silica) bilayers on a glass substrate. The primary method of generating thin films is through the ionic self-assembly of monolayers (ISAM) as presented by Iler in his 1965 paper [2]. Due to the ionic nature of the silica nanoparticles only a single layer of nanoparticles can be adhered to the slide in a single dipping. The particles adhere to the slide, and due to electrostatic repulsion, cause other particles to be repelled from the slide.

Iler discussed the application process of silica nanoparticles onto a glass substrate. For his purposes, he used a black glass substrate, which allowed him to visually inspect the slide and see the interference colors easier than a slide made of clear glass. Glass has a slight negative charge, as do silica nanoparticles, so it is necessary to apply a positively charged "glue," (Iler used alumina) to the slide before applying the nanoparticles.

In his process, he dipped a slide into a .25% alumina solution such that only half of the glass was submerged in the alumina, rinsed off the excess under running water and let it air dry. By visual inspection, there was no visible change in the surface of the slide, indicating that the alumina does not affect the appearance of the silica thin film on the surface. After, he submerged the slide into a 2% solution of aqueous silica with a diameter of 100 nm and a pH of 3, rinsed the excess off under running water and let it air dry. At this stage, when the slide is inspected visually, reflected light of a certain wavelength can be seen when looking at the slide from a glancing angle. The portion of the slide that was not submerged in the alumina had no visible film on it, because the silica particles do not attach directly to the slide. This process can then be repeated to get more layers of nanoparticles, and a different thickness of the thin film. Different thicknesses of silica thin films reflect different wavelengths of light, so by looking at the interference colors of the films, Iler was able to estimate the thickness of his films.

Building off of Iler's work, Lvov et al. [3] used a similar process to create bilayers on slides and used a quartz crystal microbalance (QCM) to obtain the film thickness. A QCM works by detecting the mass change of the layers, which leads the QCM to output a frequency change proportional to the change in mass. For their studies they used poly(diallyldimethylammonium chloride) (PDDA) as well as sodium poly(styrenesulfonate) (PSS) and poly(ethyleneimine) (PEI) dispersed in deionized water (specific resistance of $18M\Omega$ cm) at concentrations between 1.5 and 3 mg/mL as their bonding agents. The pH of these solutions was controlled using HCl or NaOH. Not only did Lvov et al. adjust the concentrations of their silica mixtures, they also adjusted the size of the particles to determine how concentration and size affects the quality of the films. The concentrations that they used were 0.1, 1, 10, 100 mg/mL at a pH of 10 and sizes of 25 ± 5, 45 ± 5 and 78 ± 5 nm in diameter. The pH of the solutions is controlled, because it has been found that at a higher pH, the quality of the samples is degraded, and the particles have a pH range at which they are stable [4].

Using the QCM, they were able to detect the change in frequency, which is linked to the change in mass during the adsorption process by the following equation: $\Delta F = (-1.83x10^8)$M/A, where M is the mass of adsorbed particles (g), A is the area of the quartz microbalance ($0.16 \pm .001cm^2$ in their case), and F is the frequency shift (Hz). They found that a change in frequency of 1 Hz is caused by a change in mass of 0.9 ng. Using this method, it is also possible to determine the film thickness with the equation: $d = 0.022(-\Delta F)$, where F is in Hz and d is the thickness of the film, in nm.

In testing, Lvov et al. attempted to adsorb SiO2 in bilayers without the use of a polymer glue in between layers. They found that the second layer of silica had a 50% mass increase from the mass of the first layer, the third had 20% of the first layer's mass, and the fourth, 5%. This decrease in mass when the silica layers are not alternated with PDDA indicates that particles are not adsorbing onto the previous bilayer. However, when silica bilayers are alternated with PDDA, a linear increase in mass is observed at each step, as outlined in Figure 1.1.

This shows not only the mass increase at each stage in the dipping process, but also highlights the difference between mass increase at different concentrations. From this image, it is evident that the PDDA deposited onto the slide at each step is relatively the same regardless of the silica concentration. In addition, the higher particle concentrations have a higher mass increase at each step. In addition, they tested the mass increase with and without rinsing the slides in deionized (DI) water, and found that there is no desorption during the rinsing process, however it does improve the likelihood of the next layer of either PDDA or silica applying evenly.

In addition to testing the mass increase of different concentrations of nanoparticles, they also tested the adsorption time at different adsorption times. Figure 1.2 shows how adsorption times

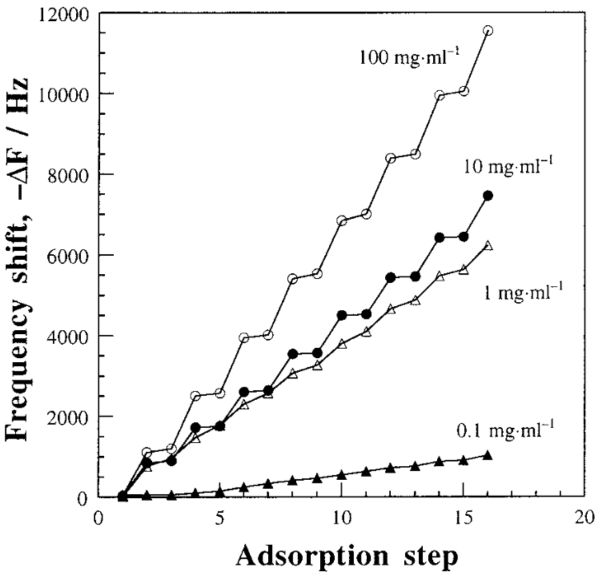


Figure 1.1: Frequency shift of QCM versus the adsoprtion step for multiple concentrations of nanoparticle solutions. The even adsorption steps account for the deposition of PDDA, while the odd numbered adsorption steps account for the deposition of silica nanoparticles. Adapted from [3].

were tested ranging from 20 minutes and decreased in time by half until they reached 15 seconds, finding that the mass increase at each time step remained constant, indicating that the deposition process occurs very rapidly.

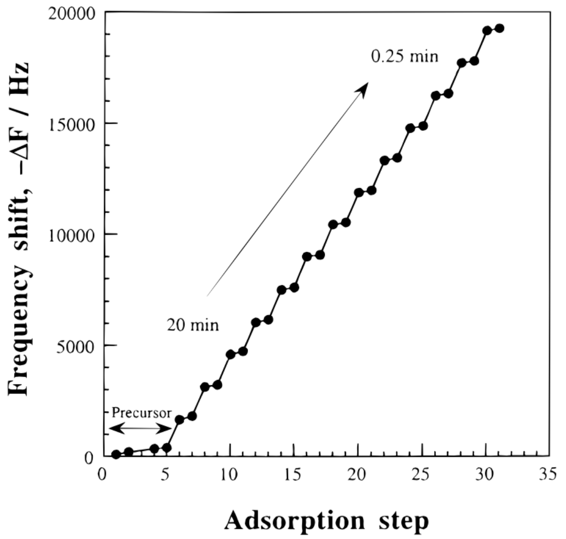


Figure 1.2: Frequency shift versus adsorption step with decreasing deposition times. The dipping time decreased in half for each adsorption step, beginning at 20 minutes, and ending at 15 seconds. Adapted from [3].

In addition to this, they discovered that drying the slide after the deposition process did not make a difference in the mass increase, and is therefore superfluous.

To get an idea of the particle clumping on the slides, Lvov et al. imaged the slides using atomic force microscopy (AFM), which shows the height profile of a section of the slide as well as a scanning electron microscope (SEM) to view the individual particles on the slide. Based on the SEM images, it was determined that the particles are closely clumped together, indicating that there is no long-range ordering of the particles. In order to more effectively estimate particle packing and the surface morphology, the AFM was used to get a height profile of the slide. Using AFM imaging it was determined that the packing coefficient of the particles was 0.56 ± 0.05, which is lower than the 0.65 that they anticipated, but still shows that void space is maintained as predicted.

The methods outlined by both Iler and Lvov et al. for deposition of particles onto a glass substrate have been used and modified in the research for this thesis. We present our deposition process in Chapter 2.

1.3 Applications of Self-Assembly

There are many wide-reaching applications for nanoparticles, but the one discussed in this thesis is anti-reflective coatings. Anti-reflective coatings on glass utilize the optical properties of light to increase transmission of light through glass, reducing reflection. When light is incident upon a surface, some of the light is reflected, and some passes through. When light hits a thin film, some of the light is reflected off of the film, and some passes through, where it hits the glass where and some light is again reflected and some passes through. This geometry can be seen in Figure 1.3.

Because of the geometry of the system, when the thickness of the film is equal to $n\lambda/4$, where n is an integer, the light undergoes a phase shift of π resulting in destructive interference of the two reflected rays. This destructive interference is therefore dependent upon the not only the thickness of the film, but also the index of refraction.

The critical index of refraction for the film to cause extinction of reflected rays is represented by $n_c = \sqrt{n_1 n_2}$, where n_1 and n_2 are the indices of refraction of air and glass, respectively. Typically, the index of refraction for air is 1, and for glass is 1.5, giving a critical index of refraction of 1.22. Obtaining an index of refraction of as close to 1.22 as possible is crucial for anti-reflective coatings. In addition, the film thickness depends which wavelengths of light are extinguished.

The final two applications which I will discuss here were both talked about by Richard Feynman in his talk at the 1959 American Physical Society meeting, and they are nanoprinting and nanocircuitry. Since Feynman's talk, there have been substantial improvements and advancements in the field of nanoprinting. Nanoprinting allows for very small devices to be printed, which is useful in engineering. It has allowed many medical devices to come to fruition, such as the aforementioned endoscopic tools. In addition, motors the size of a fingernail are now able to be printed which allows for small devices to function.

Finally, the last implication for the use of nanoscience is in nanocircuitry. As can be seen from the advances between the time of Feynman's talk, and present day, there have been great advances in the use of nanocircuitry, and electronics as a whole. In 1959, computers took up entire rooms, and now most people have a computer the size of their palm in their pocket at any given moment. Nanocircuitry has allowed for these advances due to the miniaturization that has occurred over the past 60 years, and has allowed for significant advances in quantum computing.

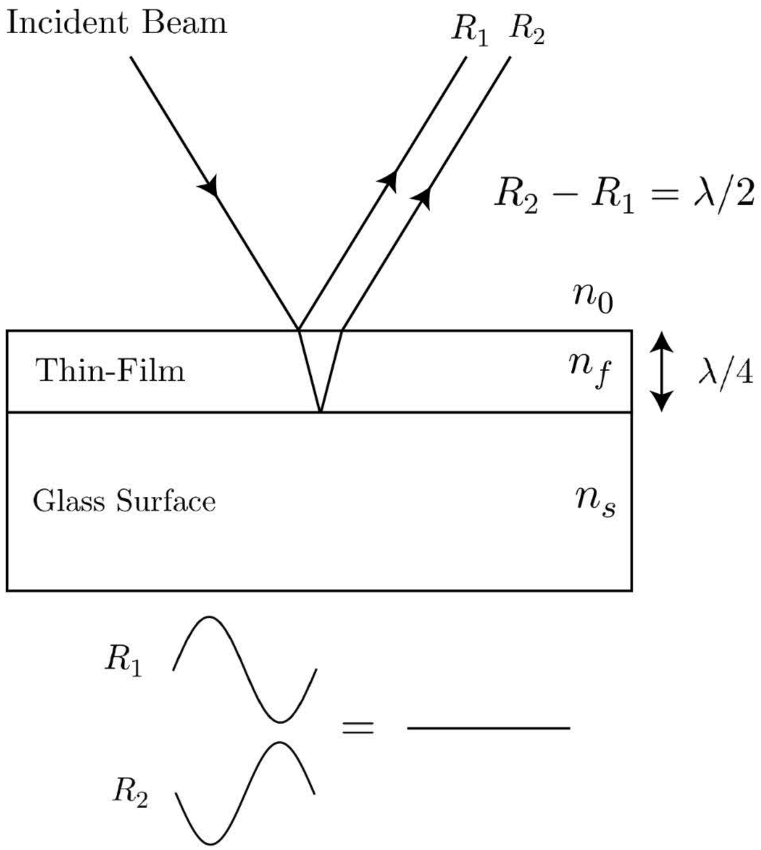


Figure 1.3: Geometry of a thin-film on a glass substrate. At each interface, a portion of the incident beam is reflected off. When the thin-film is a quarter of the wavelength of the incident light source, $R_2 - R_1 = \lambda/2$, and the two reflected waves cancel out, resulting in no visibly reflected light. Image courtesy of Matthew Withers '20.

1.4 Analysis Methods

There are several different ways view and analyze nanostructures. The primary method used for the analysis of this thesis is through the use of a Thin-Film Analyzer by Filmetrics, while the secondary method is with a Scanning Electron Microsope (SEM).

1.4.1 Thin-Film Analyzer

The primary method for imaging samples through the course of this thesis is using a thin-film analyzer by Filmetrics. The analyzer we used is in their F20 series, seen in Figure 1.4.

The thin-film analyzer measures several different properties of the thin films, such as the reflectance, transmittance, thickness, and optical constants (n and k, where n is the index of refraction and k is the extinction coefficient, both of which are dependent upon wavelength) [23]. The thin film analyzer does this by utilizing the wavelike properties of light, as well as the optical properties of thin films in order to determine these characteristics.

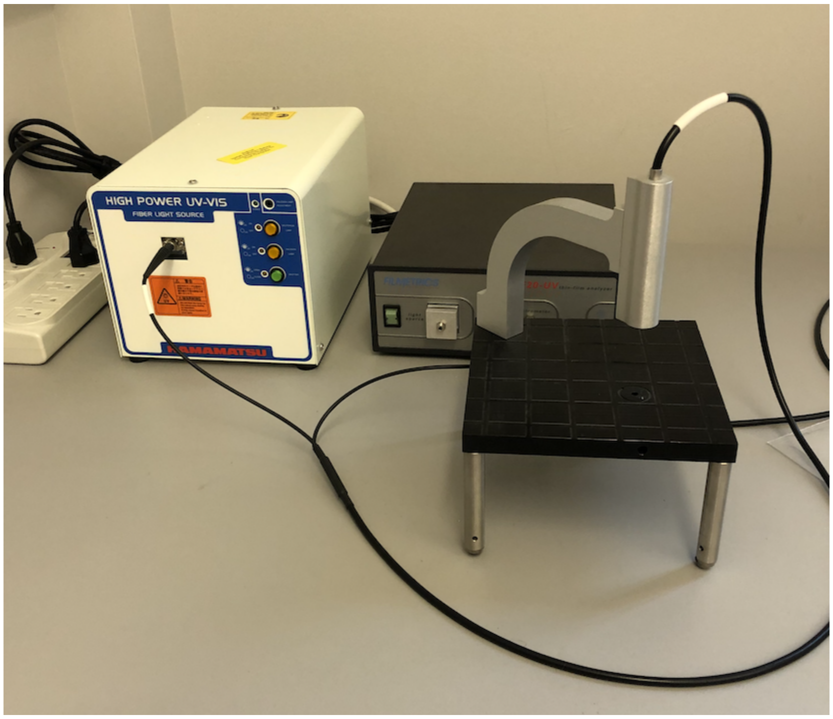


Figure 1.4: Filmetrics F20 thin-film analyzer used throughout this thesis to obtain reflectance and transmission spectra.

The thin film analyzer reflects and transmits light from a tungsten-halogen bulb which outputs light in the range from $\lambda = 375 - 3000$ nm, although the measurement system analyzes the sample between the wavelengths of 200 and 1100 nm at 1024 different wavelengths. As the light passes through the sample, the light hits both the top and the bottom surface of the thin film and reflects back light. Based on the properties of the thin film, some of the light is in phase, and constructively interferes, and some of the light is out of phase, and destructively interferes.

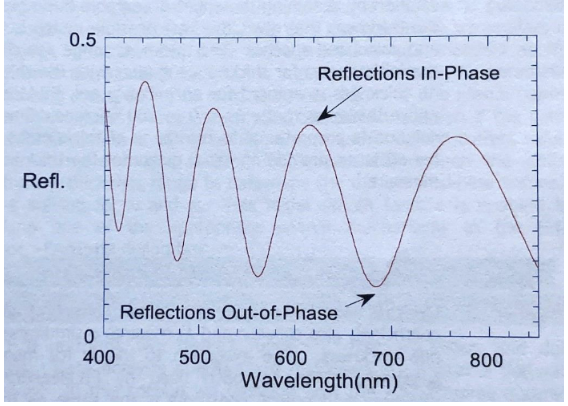


Figure 1.5: Reflectivity versus wavelength of an arbitrary sample. The oscillatory nature of the reflectivity represents the fact that there are points in which the reflections are more in-phase, or more out-of-phase, based on the geometry in Figure 1.3. Adapted from [23].

As Figure 1.5 depicts, there is an oscillatory behavior in the reflected light, and in general, the thicker the film, the more oscillations will be present within a given range.

This particular thin film analyzer measures thicknesses between 15 nm and $70\mu m$. Based on the reflectance or transmission spectra, as well as the equation, $\lambda = \frac{2nd}{i}$, where λ is the wavelength of light, n is the index of refraction, d is the thickness of the film, and i is an integer, the thickness can be determined. The user will input a film stack into the computer, telling the system how many layers of what film are on the substrate, and what substrate is used. Using this film stack, the system has a guess of what the thickness is. It inputs this thickness into the above equation and then narrows down to the correct thickness as it iterates through.

In addition to measuring the thickness, the thin-film analyzer is able to measure the optical properties, specifically the index of refraction, and extinction coefficient, both of which are dependent upon the wavelength. These two values are related by the Kramers-Kronig relation [24], which connects the real and imaginary parts of a complex function above the line $y = 0$. The Kramers-Kronig relation states:

$$\chi(\omega) = \chi_1(\omega) + i\chi_2(\omega) \tag{1.1}$$

$$\chi_1(\omega) = \frac{1}{\pi}\mathscr{P}\int_{-\infty}^{\infty} \frac{\chi_2(\omega')}{\omega' - \omega} \tag{1.2}$$

$$\chi_2(\omega) = \frac{1}{\pi}\mathscr{P}\int_{-\infty}^{\infty} \frac{\chi_1(\omega')}{\omega' - \omega} \tag{1.3}$$

where $\chi_1(\omega)$ and $\chi_2(\omega)$ are real, and $\mathscr{P}$ is the Cauchy-Principal value. Due to the fact that these relations occur over the full range, and we only have values from 200-1100 nm, we cannot directly use these relations to find k from n, or vice versa. Since $n(\lambda)$ is related to $n(\lambda \pm 1)$, we can find the relationship between all of the values of n, and using the thickness calculated by the instrument can find the equation for n. Once n has been found, using the Kramers-Kronig relations, k can be found for that section of the spectrum.

On a deeper level, the way that the thin film analyzer works is as follows. The reflected or transmitted light is collected by an optical fiber cable and a lens and the intensity of the reflected light is measured by a spectrometer. Within the system lies a diffraction grating which disperses light, which is then captured by a linear photodiode array. There are 1024 linear photodiodes in this array, each corresponding to a certain wavelength. The linear photodiodes trigger a current to flow, and the amount of current flow at that wavelength is proportional to either the transmittance or reflectance. In order to decrease the interference from other sources in the system, the measurements are taken in a dark room, and there is a baseline measurement taken to account for the response from the machine. This baseline measurement takes a sample of the background noise as well as from a reference sample to calibrate the machine.

1.4.2 Scanning Electron Microscopy (SEM)

The idea for a scanning electron microscope (SEM) was first introduced by Ernst Ruska, a German physicist, in 1927 [9]. He discovered that using electron ray bundles, he could create a microscope with a varying focal length to image and magnify samples. Using a varying current, he was able to change the focal length of the lens to image a more diverse array of samples. He also hypothesized that with combinations of these lenses, that he could increase the magnification of samples even further. This concept was developed further, and in 1931, Ruska proved that he could resolve images at up to 12,000 times magnification through the use of an electron beam.

In order to generate images from an electron beam, there needs to be a beam source to produce a stream of electrons [1]. The most commonly used electron source is a thermionic gun, which is a filament coated with a metal with a low work function. This filament is then heated to a very high temperature, until electrons are thermally excited out of the metal, and then accelerated towards the sample. This relies on the thermionic current density, J, which states:

$$J = \frac{4\pi me}{h^3} k_B T^2 exp \frac{\phi}{k_B T} \tag{1.4}$$

where ϕ is the work function of the metal, m is the mass of an electron, e is the charge of an electron, h is Planck's constant, k_B is Boltzmann's constant, and T is the temperature of the filament.

After the electrons leave the beam, they pass through a condenser lens which collimates the beam such that the electrons are focused on the focal point of a lens, which in turn forms a small spot on the sample. This spot is scanned over the sample in a raster pattern (moves side to side as it scans down the sample), and the interaction between the electrons and the sample is collected by a detector. For thin samples, the detector can be placed below the sample, in which the image is formed due to the transmitted current. For thicker samples, the detector is placed above the sample, and it collects the backscattered electrons in order to form an image of the sample.

To focus and resolve the beam, the de Broglie equation can be utilized to calculate the resolution of the electron microscope. The de Broglie equation states $\lambda = h/\sqrt{meV}$, where V is the energy of the electron in volts. Based on this equation, if V is only 5 kV, the wavelength of the electrons is 0.17 $\AA$, meaning it can resolve images down to $0.17 * 10^{-10}$ meters in size. Due to the small resolution size, scanning electron microscopy is a very useful tool for imaging nanoparticles. That being said, in order to preserve the sample, as well as to make it conductive so that the electrons can interact with the nanoparticles, it is necessary to gold coat the sample. This adds to the time consuming nature of the process of imaging samples under an SEM, making it a secondary imaging technique for the purposes of this thesis.

Today, an SEM looks like what is shown in Figure 1.6, and produces images like the one in Figure 1.7.

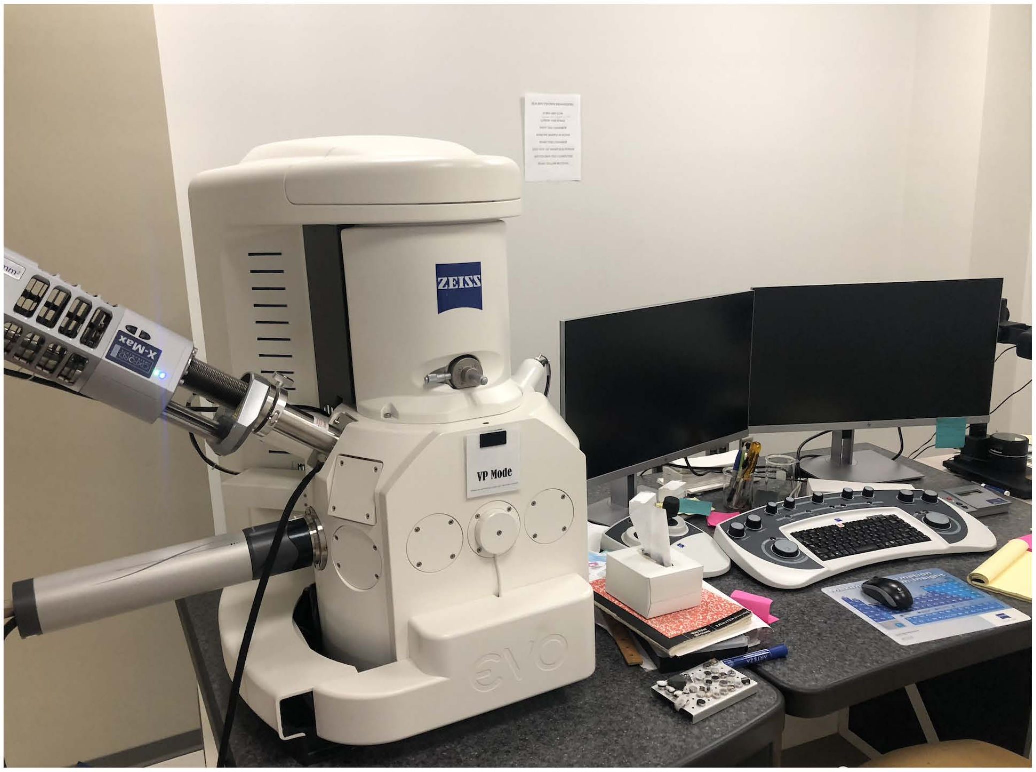


Figure 1.6: Set-up of an SEM. The sample is placed inside the chamber seen on the left-hand side of the photo, and the image can be seen on the monitors seen on the right-hand side of the photo. The image is focused, zoomed in, and adjusted using the knobs on the keyboard.

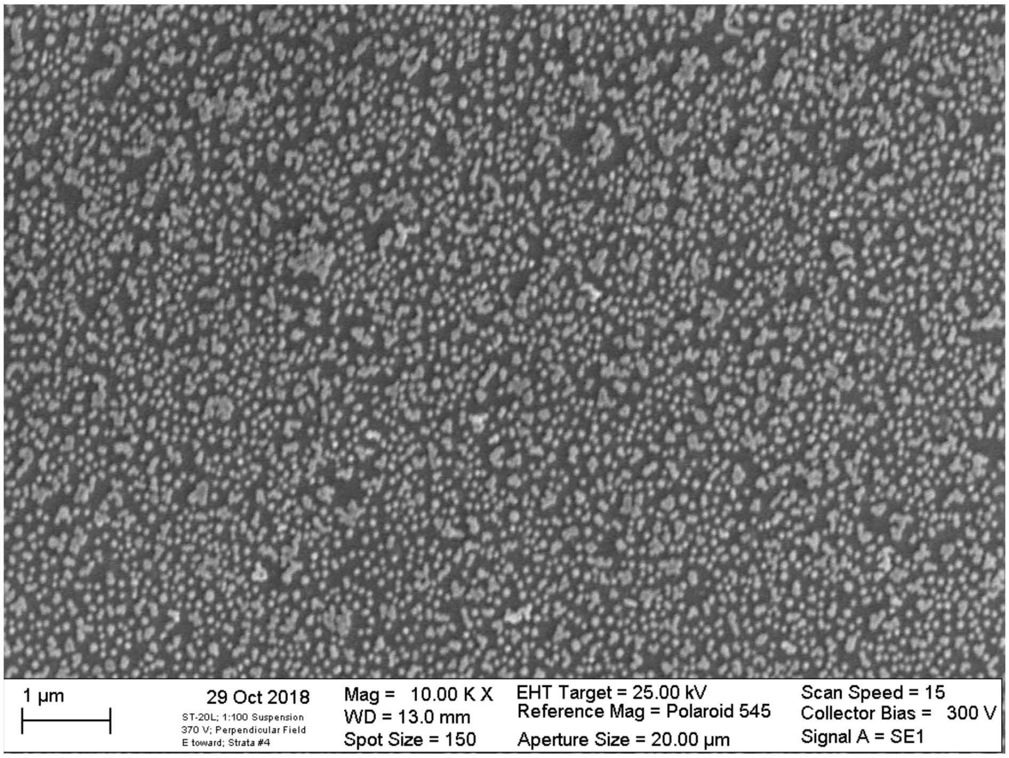


Figure 1.7: Sample SEM image of silica nanoparticles on a glass substrate.

Chapter 2

Layer-by-Layer Self-Assembled Silica Nanoparticle Thin Films

2.1 ISAM Overview

In order to compare theoretical models to experimental data, it is necessary to meticulously develop samples in order to image them. To begin, clean microscope slides are visually inspected to ensure there are no impurities on the slide. Using a sonicator bath filled with water, glass slides are placed into a container filled with acetone and sonicated for five minutes. This process is repeated with containers filled with methanol and then isopropyl alcohol, before being placed in a container of de-ionized water (DI water), where the slides remain until being dried under a flow of nitrogen gas.

Using the clean glass slides, we employ the process of ionic self-assembly of monolayers (ISAM) to develop uniform layers of nanoparticles. The process in Figure 2.1 is used to deposit the silica nanoparticles onto the substrate.

Glass slides have a slight negative charge, as do the silica nanoparticles, so to deposit the nanoparticles onto the slide [11], a layer of polydiallylmethylammonium chloride (PDDA) is deposited onto the slide. The purpose of using PDDA, which is a cationic polymer, is to act as a glue for the particles. It is important to note that the PDDA does not affect the optical properties of the thin film. The PDDA has positively charged molecules which form a polyelectrolyte (a polymer made up of macromolecules with a high proportion of ionic or anionic charge) [25] which adhere to the negatively charged glass slide, and allow the negatively charged silica particles to adhere to the substrate. The glass slide then sits in a staining jar filled with a 1 mM solution of PDDA for three minutes and is then rinsed in a container of DI water. After, the slide is placed in a glass staining jar with a concentration of 100:1 DI water to silica nanoparticles (we used SNOWTEX ST-20L silica nanoparticles by Nissan Chemicals, which have a diameter of 40-50 nm) where it sits for three minutes before being dried under N_2 gas. We refer to the layer of PDDA and the layer of nanoparticles as a bilayer. This process may be repeated to obtain multiple bilayers.

Through exploratory experiments in the summer of 2017, it was determined that particle deposition occurs very rapidly, and a steady state of particle coverage density is rapidly achieved. That being said, the slide remains in the silica nanoparticles for the full three minutes to ensure full coverage [14]. If the process is to be repeated for a high number of bilayers, a dipping machine (we use the StratoSequence VI by nanoStrata Inc.) is used. This machine (seen in Figure 2.2) allows

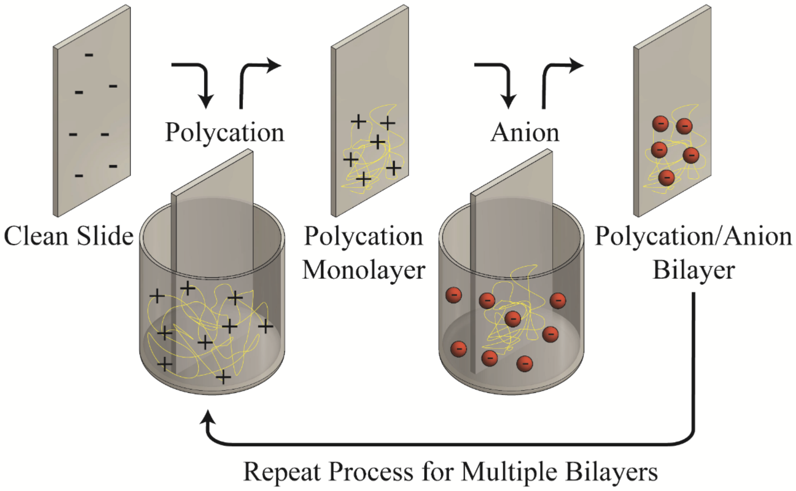


Figure 2.1: A clean slide with slight negative charge is dipped into PDDA with slight positive charge, giving the slide a positive charge. Then the slide is dipped into the anionic nanoparticle suspension, giving the slide a negative charge. This creates a single bilayer of nanoparticles, and the process can be repeated to obtain multiple bilayers. Reproduced from [14] and [26].

eight containers to be held at the same time with four slides being dipped at a given time. The machine can be set to deposit any desired number of bilayers onto the slide.

Once the samples have been produced, there are multiple ways to image the slides. The first, most time-consuming method involves using a scanning electron microscope (SEM) to get an image of what the particles look like on the slide. Because this method is very time consuming, it is by far the least effective method. In addition, this method doesn't give any information regarding the optical properties of the thin-film, simply the particle density on the slide. The method which has proven to be the most useful for obtaining data from the samples is through the use of a thin-film analyzer, by Filmetrics, which is therefore the primary analysis method used in this thesis.

2.2 Optical Properties of Thin Films

In order to determine the index of refraction, n, of our samples, we use a method presented by Swanapoel in [21]. This method uses data from the fringes of the transmission spectra in order to accomplish this. In these calculations, s is the index of refraction of the substrate with no film on it, and is calculated with the equation:

$$s = \frac{1}{T_s} + (\frac{1}{T_s^2} - 1)^2 \tag{2.1}$$

where T_s is the transmission of the thick substrate. For our calculations, we find T_s to be .920.

Swanapoel in his paper uses the fringes of the transmission spectrum directly for the calculations to find the index of refraction. However, our data shows fringes for the reflectance, but not the transmission spectrum. This occurs because the Filmetrics instrument measures the transmission

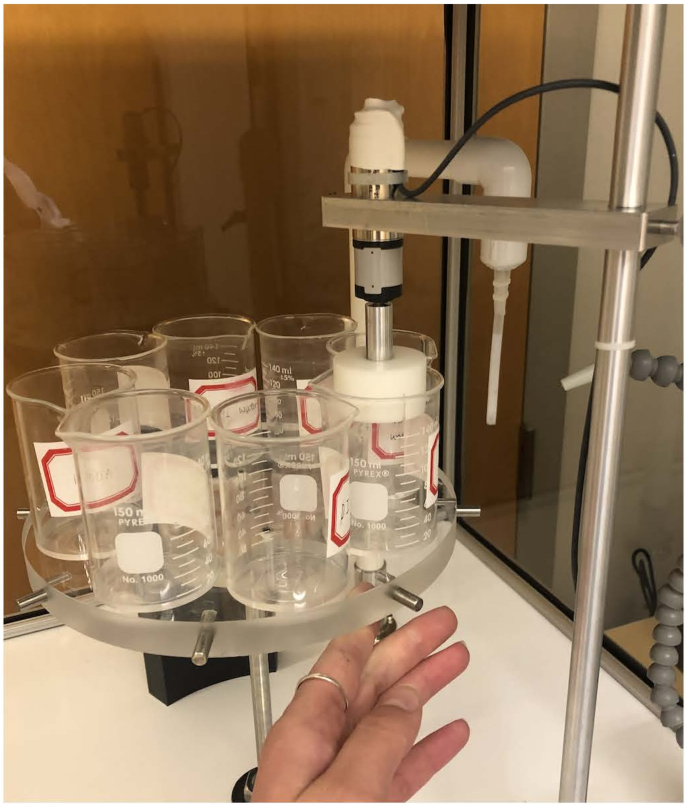


Figure 2.2: This image shows our dipping machine, StratoSequence VI by nanoStrata Inc. in use. The jars are oriented in the machine such that they are dipped in PDDA, rinsed in DI water, dipped in nanoparticles, and rinsed in DI water again.

through the entire sample, which is the transmission through not only the thin film, but also the slide. To find the transmission of the thin film only, we subtract our reflectance spectrum from 1 to obtain transmission as a function of wavelength. We are able to do this because our extinction coefficients were found to be very low (less than 0.02 for all samples, when they are normalized to 1), indicating that there is little extinction within the sample. The extinction coefficient is calculated as $1 - R - T$, where R is the reflectance of the sample, and T is the transmission of the sample.

By plotting the transmission as a function of wavelength, we are now able to find the index of refraction of each of the samples by looking at the maxima and a minima of the fringes. Swanapoel found an equation for the curve that follows the tops of all of the maxima, T_M, as well as an equation for the curve that follows all of the bottoms of the minima, T_m. They were able to find these equations, because they had several fringes, whereas four our samples we typically only observe one or two. Swanapoel's data can be seen in Figure 2.3, where our data can be seen in Figure 2.4.

Using the data presented in Figure 2.4, we are able to find the maxima and minima of one of the fringes, and use these values as our T_M and T_m values. The downside to this is that we are only getting a single value for the index of refraction of the slide, as opposed to the index of refraction of the slide as a function of wavelength, like the Swanapoel paper presents. However, we are confident that dispersion is minimal with our samples so this is not as big of an issue as it

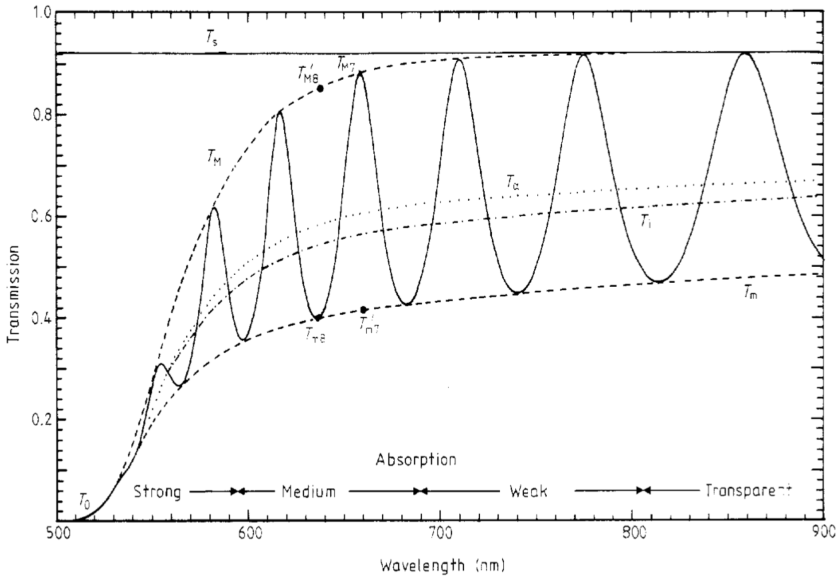


Figure 2.3: Transmission spectrum from [21]. The top dotted line shows the line T_M, the bottom dotted line depicts the line T_m, and the solid line is their transmission data. Reproduced from [21].

might initially seem.

Using these values, Swanapoel presents several classes of equations for different regions of the wavelength spectrum. It is not evident from this paper how the classification between the strong, medium, weak and transparent absorption regions is defined. The first region presented is that of the transparent region, where the absorption coefficent, α, is zero. Using this method, only the minima of the transmission is required, and the index of refraction is calculated with the following two equations:

$$M = \frac{2s}{T_m} - \frac{s^2+1}{2} \tag{2.2}$$

$$n = (M + (M^2 - s^2)^{1/2})^{1/2} \tag{2.3}$$

Similarly, the index of refraction in the weak and medium absorption regions can be calculated. In these regions, $\alpha \neq 0$, and both T_M and T_m are required to find the index of refraction. The index of refraction can be calculated with the following two equations:

$$N = 2s\frac{T_M - T_m}{T_M T_m} + \frac{s^2+1}{2} \tag{2.4}$$

$$n = (N + (N^2 - s^2)^{1/2})^{1/2} \tag{2.5}$$

For the region of strong absorption, there are no interference fringes, so there is no way to calculate the index of refraction for this region.

Swanapoel goes on to explain how to determine the thickness of the films using the index of refraction of two adjacent maxima or minima. For this calculation, n_1 is the index of refraction at

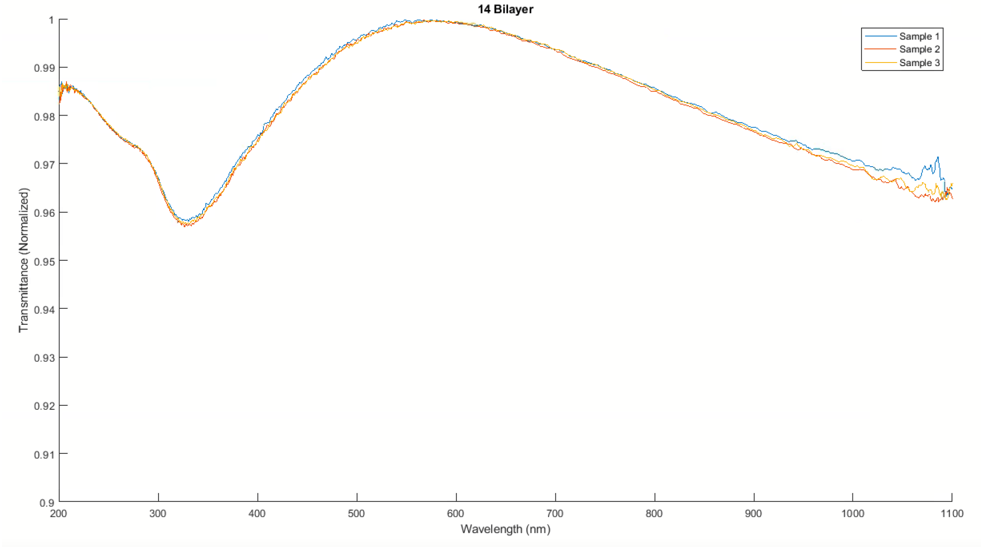


Figure 2.4: Transmission data obtained from a slide with 14 bilayers of silica nanoparticles at a concentration of 100:1 de-ionized water:silica nanoparticles.

the maxima or minima with corresponding wavelength, λ_1, and n_2 is the index of refraction of the adjacent maxima or minima at $\lambda = \lambda_2$. With these values, the thickness can be determined with the equation,

$$d = \frac{\lambda_1 \lambda_2}{2(\lambda_1 n_2 - \lambda_2 n_1)} \tag{2.6}$$

Swanapoel notes that this method of determining the thickness of a sample is very sensitive to errors in the calculation of the index of refraction, and is therefore not very accurate. Alternatively, there is the equation $2nd = m\lambda$, which determines the order (m) of the interference fringes, where n is the index of refraction, d is the thickness of the sample, m is an integer for maxima or a half integer for minima, and λ is the wavelength. Rewriting this equation, we can obtain the equation for a straight line, $l/2 = 2d(n/\lambda) - m_1$, where l is an integer, and m_1 is the order of the first extreme. By plotting $l/2$ versus n/λ, a straight line will be formed, and the slope will be equal to twice the thickness, allowing us to determine the thickness of the film.

With the thickness as well as the index of refraction of the film known, we are now able to determine x, which is an intermediate step to find the absorption coefficient of the film. There are several different equations to determine x, but Swanapoel found that the method which gave the most accurate results for α was through using the following two equations:

$$E_M = \frac{8n^2 s}{T_M} + (n^2 - 1)(n^2 - s^2) \tag{2.7}$$

$$x = \frac{E_M - (E_M^2 - (n^2-1)^3(n^2 - s^4))^{1/2}}{(n-1)^3(n-s^2)} \tag{2.8}$$

Once x has been calculated, we can take the relation $x = exp(-\alpha d)$ in order to find α. We can take the natural logarithm and rearrange in order to find that $\alpha = -\frac{ln(x)}{d}$ and solve for α.

To find the last optical constant, k, which is the extinction coefficient, the equation $k = \frac{\alpha\lambda}{4\pi}$ can be used. Swanapoel presents a series of equations which allow us to calculate all of the optical constants of each thin film through just the transmission spectrum of the film.

2.3 Experimental Results

Using the technique outlined in Figure 2.1, we generate samples ranging in number of bilayers from zero to eighteen utilizing the dipping machine. With these samples, we measure the reflectances and transmittances at five spots on each sample. This is done by outlining a slide on a piece of graph paper, and outlining five boxes on the paper. These are then traced onto the slide, to ensure that we are imaging the reflectance and transmittance in the same location for each sample.

With each of these sets of data, we employ the method presented in Section 2.2 to calculate the index of refraction. Due to a lack of fringes in the transmittance spectra for the number of bilayers ranging from zero to six, we were unable to calculate the index of refraction for these samples. The results of our calculations may be seen in Figure 2.5.

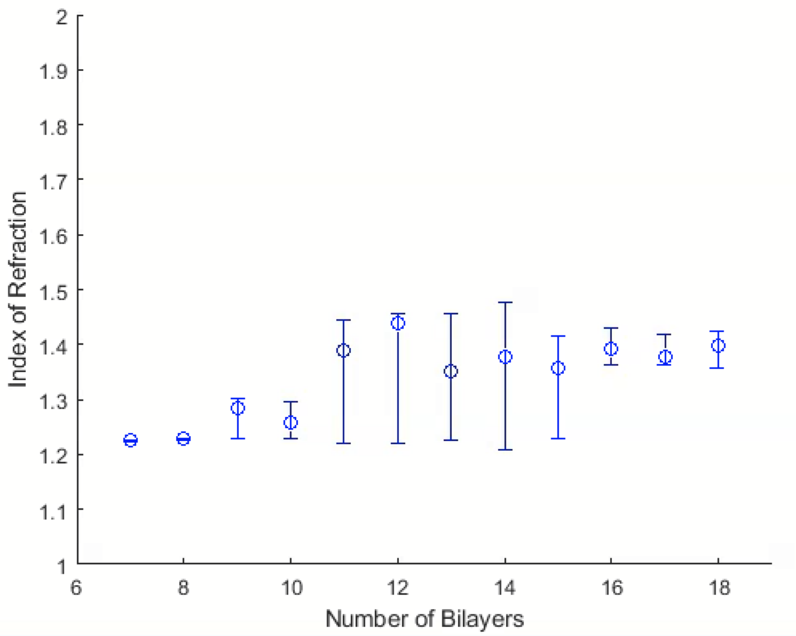


Figure 2.5: This graph shows the index of refraction as a function of the number of bilayers for seven to eighteen bilayers. The error bars were made by taking the highest and lowest index of refraction for a given number of bilayers and having those be the top and bottom of the error bar. The most central value for the index of refraction is the data point that is plotted on the graph.

The different lengths of the error bars in Figure 2.5 arises from the fact that there were significantly more samples taken in the eleven to fourteen bilayer range. In addition, it should be noted that the index of refraction ranges from around 1.2 to 1.4, which is around the critical index of refraction of 1.22, that was calculated in Section 2.2.

In addition, we are able to obtain a correlation coefficient for these values to tell us how correlated the index of refraction and number of bilayers are. The correlation coefficient ranges from -1 to 1, and tells us how linear the results are. A value of 1 indicates perfect (positively) linear correlation, 0 indicates no correlation, and a value of -1 indicates a perfect (negatively) linear correlation. Using the MATLAB correlation function, we obtain a correlation value of .766, indicating very good correlation between the number of bilayers and the index of refraction.

2.4 Simulation Results

While creating samples, and analyzing them under a thin-film analyzer is not generally a very time-consuming process, it is helpful to have computer simulations to predict the behavior, and optical properties of the thin-films before they are produced. Through computer simulations, we are able to determine the number of bilayers which should be sampled in order to create a thin-film with a particular index of refraction, and thickness.

A common way to model physical systems such as this is through the use of a cooperative sequential adsorption with evaporation (CSAE) model [12]. These models account for electrostatic interactions between particles throughout the deposition process, as well as allows for evaporation of particles from the slide. The type of CSAE model we use is called a CSAE-NN model, where the NN stands for "nearest neighbors" because the model accounts for the presence of other particles directly next to other spots on the lattice. The geometry of the particles on a glass slide, with the uniform PDDA layer allows us to model this as a stack of two-dimensional lattices. Therefore, our model is defined on a three-dimensional square lattice. In this model, for each layer of the system, each site, i, on the lattice may either be filled, corresponding with an occupation number $n_i = 1$, or unfilled, with $n_i = 0$. An example lattice may be seen in Figure 2.6.

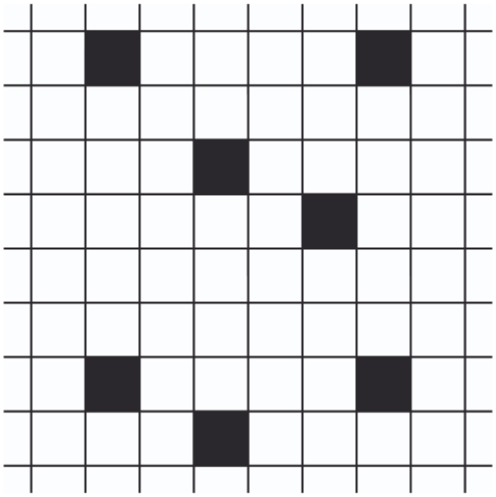


Figure 2.6: Example lattice similar to the ones used in our models. Here a black square corresponds to an occupied site, and a white square corresponds to an unoccupied site. Reproduced from [26].

To effectively model this under different parameters, a transition rate must be defined. For this system, the transition rate is defined as:

$$c(n_i \to (1 - n_i)) = (1 - n_i)(\alpha_1 \beta_1^{\eta_1} + \alpha_2 \beta_2^{\eta_2} + \gamma(n_i) \tag{2.9}$$

where $\eta_1 = \sum_{j \in NN} n_j$, and $\eta_2 = \sum_{k \in NN} n_k$. Here α_1 determines the concentration of the reservoir, β_1 defines the electrostatic screening between particles within a layer, and the η_1 term determines the number of nearest neighbors of site i within a layer. Similarly, α_2 determines the bulk concentration, β_2 determines the electrostatic screening of particles between layers, and the η_2 term determines the number of nearest neighbors of site i between layers. Finally, γ accounts for particle detachment from the slide due to evaporation. The values of each of these constants can be in the range from zero to one.

Because of the multilayer system, the β and η terms may vary depending on the location within the lattice, and the presence of nearby particles. For example, η_1 accounts for the number of nearest neighbors within a layer, and η_2 accounts for the nearest neighbors in between layers. A particle neither on the top or bottom layer will have six nearest neighbors (four within the layer, one above, and one below), while a particle on the top or bottom layer will have five nearest neighbors (four within the layer, and one either above or below). Based on the concentration of particles in the system, the β terms will vary.

Due to the nearest neighbor interactions that arise from the CSAE-NN models, the system is very difficult to solve with generic analysis methods. We employ an approximation used in statistical physics called *mean field theory* [26]. This replaces the local "field" generated by the interaction from the neighbors with a mean field, which is averaged over the lattice. This approximation assumes a uniform distribution of particles over the slide, and that each side will have approximately the same number of nearest neighbors as every other site, meaning each site feels the same effect. Because of the large lattice size compared to the size of the particles, edge effects are negligible. We therefore find that the particle density is uniform across the slide.

The following derived equations may be applied to other multilayer systems for different numbers of nearest neighbors. This allows us to consider other geometries such as multilayered Cayley-trees, though the following equations focus on the multilayered lattices. We therefore derive a time-dependent equation for the particle density.

$$\frac{\partial \langle n_i \rangle}{\partial t} = (1 - n_i)(\alpha_1 \beta_1^{\eta_1} + \alpha_2 \beta_2^{\eta_2}) - \gamma \langle n_i \rangle \tag{2.10}$$

Neighboring sites are uncorrelated mathematically, so the ensemble average of nearest-neighbors is approximated as the product of the mean site occupations:

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

With this approximation, we obtain the following equation for the time-dependent average site occupation:

$$\frac{\partial \langle n_i \rangle}{\partial t} = (1 - \langle n_i \rangle)(\alpha_1 \beta_1^{\langle \eta_1 \rangle} + \alpha_2 \beta_2^{\langle \eta_2 \rangle}) - \gamma \langle n_i \rangle \tag{2.12}$$

with $\langle \eta_1 \rangle = \sum_{j \in NN} \langle n_j \rangle$, and $\langle \eta_2 \rangle = \sum_{j \in NN} \langle n_k \rangle$, the sum of the nearest neighbors of site i in the j, and k directions, respectively.

The rate equation (2.12) describes the change of the average occupation of a site, i with time. The loss term (γ) represents the evaporation of a particle at site i, and the gain terms (α) for deposition of a particle, if i is empty. The deposition is dependent on the interactions between particles, as well as the number of occupied neighbors.

In the approximation above, we stated that $\langle\beta^{\eta}\rangle = \beta^{\langle\eta\rangle}$, which in general is not true, though in a first approximation, we make this assumption because:

$$\begin{aligned}
\langle\beta^{\eta}\rangle = \langle e^{\eta\ln\beta}\rangle =& \langle 1 + \eta\ln\beta + \frac{1}{2}\left(\eta\ln\beta\right)^2 + \dots\rangle \\
=& 1 + \langle\eta\rangle\ln\beta + \frac{1}{2}\langle\eta^2\rangle\ln^2\beta + \dots \\
=& 1 + \langle\eta\rangle\ln\beta + \frac{1}{2}\langle\eta\rangle^2\ln^2\beta + \dots \\
=& e^{\langle\eta\rangle\ln\beta} \\
=& \beta^{\langle\eta\rangle}
\end{aligned}$$

To be thorough, one is supposed to write rate equations, as seen in (2.12) for all N sites of the lattice. To simplify this problem, we assume translational invariance, allowing us to remove the location dependence from site averages.

$$\begin{aligned}
\langle n_i\rangle &= \langle n\rangle \\
\langle\eta\rangle &= z\langle n\rangle
\end{aligned}$$

Here z is the *mean* number of nearest-neighbors of each site. This approximation is valid for systems in which the coverage density is mostly uniform, and edge effects are negligible.

With this, the particle density is defined by:

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

where N is the number of sites on the lattice. This gives us a rate equation for particle density of:

$$\frac{\partial\rho}{\partial t} = -\gamma\rho + (1-\rho)(\alpha_1\beta_1^{z_1\rho} + \alpha_2\beta_2^{z_2\rho}) \tag{2.13}$$

where z_1 accounts for the number of neighbors within the layers, and z_2 accounts for the number of neighbors between layers.

This equation may be solved graphically using computer software such as Python or Maple. Although this equation is much more accessible than (2.12), information such as particle correlations and probability distributions is lost.

With Python, we are able to generate graphs representing the particle density as a function of time for various values for each parameter.

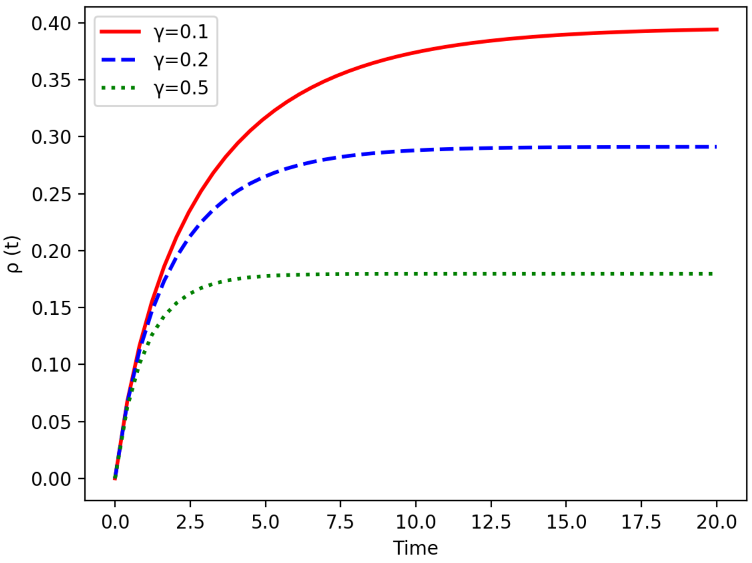


Figure 2.7: Particle density as a function of time for the following parameters: $\alpha_1 = 1$; $\alpha_2 = 0.1$; $\beta_1 = 0.2$; $\beta_2 = 0.5$, with three different values for γ. γ determines the evaporation of particles from the slide. As anticipated, as gamma increased, the particle density decreases.

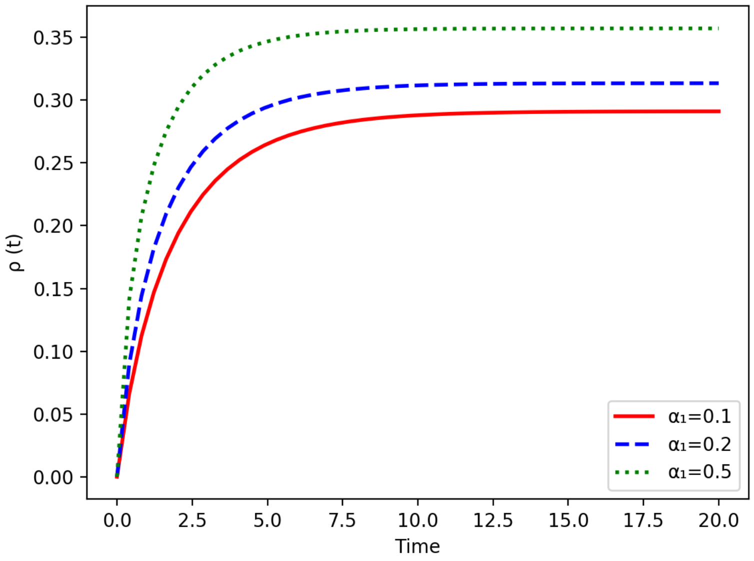


Figure 2.8: Particle density as a function of time for the following parameters: $\gamma = 0.2$; $\alpha_2 = 0.1$; $\beta_1 = 0.2$; $\beta_2 = 0.5$, with three different values of α_1. α_1 accounts for the reservoir concentration. As the concentration of the reservoir increases, the particle density increases as well, in concordance with our expectations.

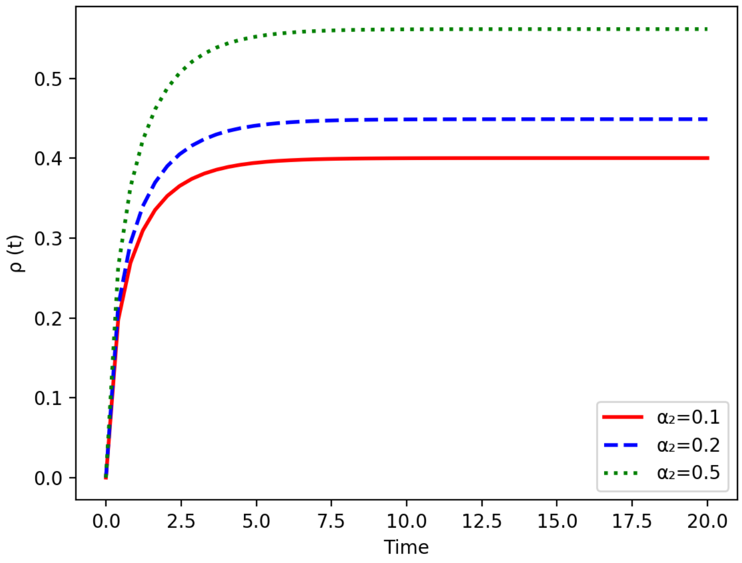


Figure 2.9: Particle density as a function of time for the following parameters: $\gamma = 0.2$; $\alpha_1 = 1$; $\beta_1 = 0.2$; $\beta_2 = 0.5$, with three different values of α_2. α_2 determines the bulk concentration of the system. As the concentration increases, so does the particle density, which is the relationship that we anticipate.

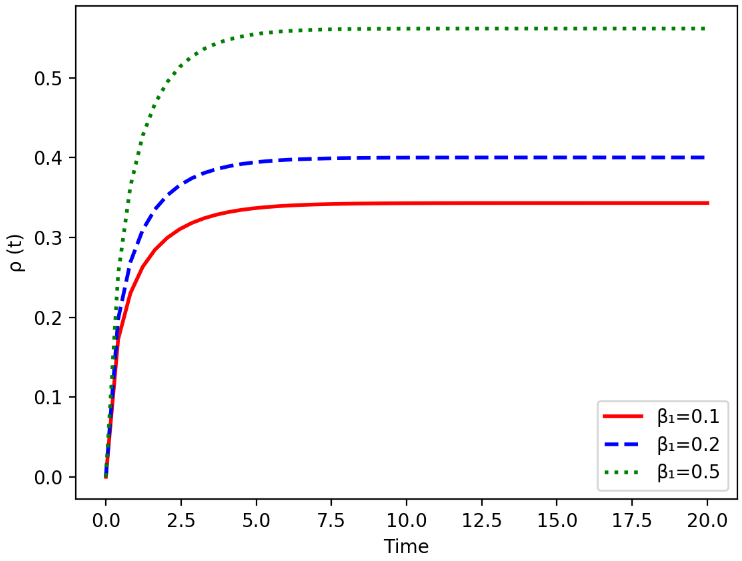


Figure 2.10: Particle density as a function of time for the following parameters: $\gamma = 0.2$; $\alpha_1 = 1$; $\alpha_2 = 0.1$; $\beta_2 = 0.5$, with three different values of β_1. β_1 represents the electrostatic screening between particles in the same layer. As expected, with increasing screening (i.e. decreasing interaction between the particles), the particle density increases.

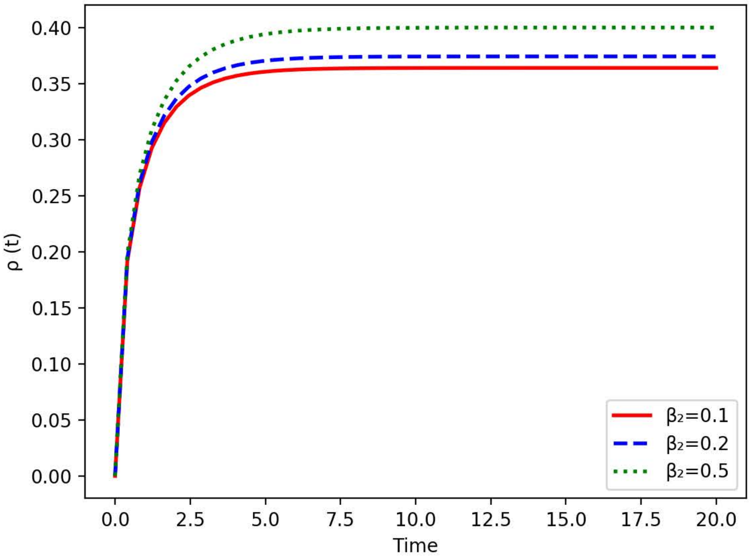
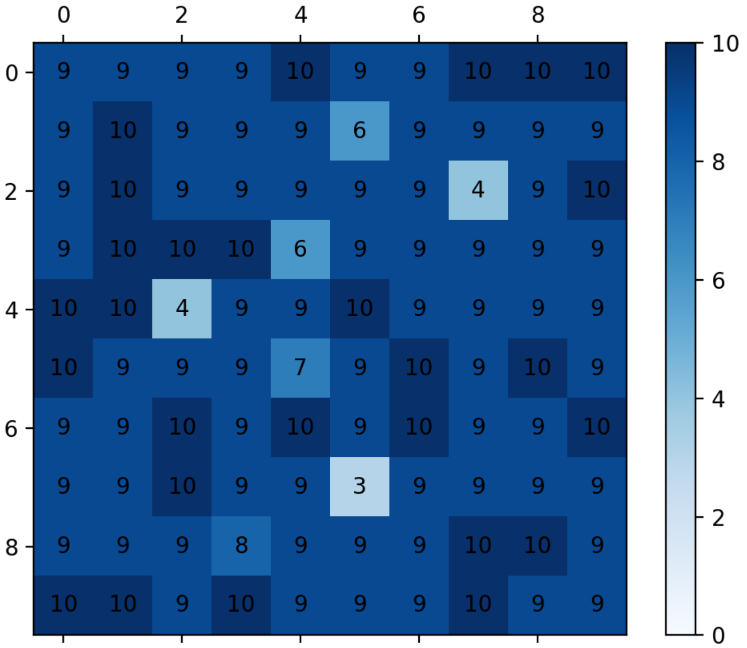


Figure 2.11: Particle density as a function of time for the following parameters: $\gamma = 0.2$; $\alpha_1 = 1$; $\alpha_2 = 0.1$; $\beta_2 = 0.5$, with three different values of β_2. β_2 determines the electrostatic screening between particles in neighboring layers. This term has a minimal effect on the outcome of the particle density, because this term is raised to the number of layers, which is only two (if a particle exists both above and below that spot on the lattice). Because of this, there is little difference for varying values of β_2, though higher values for this screening coefficient will lead to higher particle densities.

The graphs presented above allow us to visualize how the different parameters affect the particle density, as well as to confirm our expectations of the transition rate. As outlined above, our expectations were confirmed, and the results of these graphs are not surprising.

In addition to being able to determine the particle density as a function of time, we are also able to output a sample lattice, which shows the number of particles occupying each site on the lattice. The purpose of this is to be able to better visualize what is physically happening on the lattice. An example of this may be seen in Figure 2.12.

Graphs of the particle density over time based on the given parameters, as well as the number of bilayers are generated. From there, Figure 2.13 was generated by collecting data from each of the particle density over time graphs as a function of time. Because the particle density saturates very rapidly, the second half of the data for $\rho(t)$ is fairly constant, so by averaging the second half of the data set, we can determine the overall particle density for the sample. Collecting this data for a 2 bilayer sample, all the way to a 20 bilayer sample, allowed us to create the graph seen above.



	0		2		4		6		8	
0	9	9	9	9	10	9	9	10	10	10
	9	10	9	9	9	6	9	9	9	9
2	9	10	9	9	9	9	9	4	9	10
	9	10	10	10	6	9	9	9	9	9
4	10	10	4	9	9	10	9	9	9	9
	10	9	9	9	7	9	10	9	10	9
6	9	9	10	9	10	9	10	9	9	10
	9	9	10	9	9	3	9	9	9	9
8	9	9	9	8	9	9	9	10	10	9
	10	10	9	10	9	9	9	10	9	9

Figure 2.12: This shows an example 10 bilayer lattice with the number of nanoparticles stacked on each site on the lattice. The darker colors indicate more particles in a given site, while the lighter colors indicate the opposite

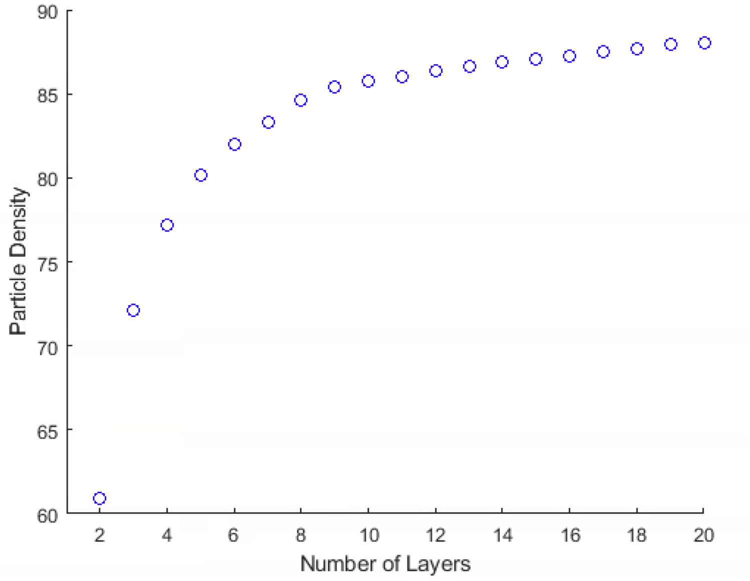
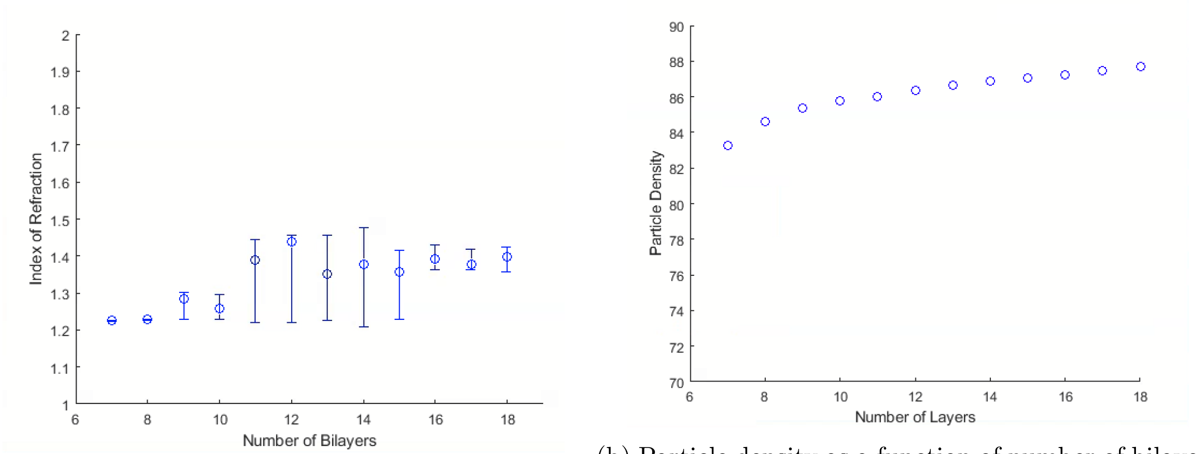


Figure 2.13: This shows the Python simulated results of particle density as a function of number of bilayers for 2 all the way up to 20 bilayers. As expected, the particle density increases with the number of layers, and eventually flattens out, when it reaches a “saturation” region.

2.5 Discussion

For ease of comparison, we present again the graph of the index of refraction vs. the number of bilayers shown in Figure 2.5, this time side-by-side with a portion of the graph of the particle density vs. the number of bilayers shown in Figure 2.13. The two graphs may be seen in Figure 2.14.



(a) The graph presented in Figure 2.5

(b) Particle density as a function of number of bilayers from 7 bilayers to 18 bilayers.

Figure 2.14: The index of refraction as a function of number of bilayers compared to the particle density as a function of number of bilayers.

To better compare the simulation results with the experimental data, we present the particle density as a function of the number of bilayers in the same range that we have data for the index of refraction (7 to 18 bilayers). For this range of particle density, we obtain a value for the correlation coefficient of .945, indicating a nearly perfect linear relationship between the number of bilayers and the particle density. Additionally, we can find a correlation coefficient for the relationship between the index of refraction and the particle density to determine how closely correlated those two values are. We find a correlation coefficient of .815, indicating that the model closely resembles the experimental data.

Though the correlation is not quite perfect, it describes the relationship like we would expect it to. As can be seen from Figure 2.14b, we are in the saturation region of the particle density versus number of bilayers, so the difference in particle density does not vary greatly from bilayer to bilayer. We see a similar relationship between the index of refraction and the number of bilayers. The index of refraction does not vary greatly from bilayer to bilayer. Because of the good correlation between the index of refraction and the particle density, we determine that the computer simulations can fairly accurately predict the outcome of the index of refraction of the samples before they are produced.

Chapter 3

Exploratory Experiments

"Research is the process of going up alleys to see if they're blind." - Barstow Bates [33].

Throughout the course of conducting research for this thesis, there have been many exploratory experiments which have helped us narrow in on the focus of optical properties of thin films. The first experiments that were performed investigated the effects of external fields on the particle density of the thin films. In particular, the focus was on electric fields oriented perpendicular to the glass substrate. The second set of experiments were performed revolved around using titania (TiO_2) nanoparticles as opposed to silica nanoparticles. Both of these exploratory experiments will be discussed in detail below.

3.1 Perpendicular Electric Fields

Due to the negative electrical charge of silica nanoparticles, it is hypothesized that through the use of external fields, we should be able to control the particle density of the silica nanoparticles on the glass substrate [5]. In order to model this scenario, we use the cooperative sequential adsoprtion with evaporation (CSAE) model [12]. This model accounts for the interactions between the nearest neighbors throughout the deposition process, and allows for evaporation as well. This method works for constant and variable fields, and using the mean field approximation, we are able to determine rate equations for the particle density, ρ. This allows us to predict the change in particle density over time, as well as to determine a final steady state particle density.

The models presented below account for both steady state, as well as time-dependent fields [15]. However, throughout the process of performing experiments, we focused primarily on constant electric fields oriented perpendicular to the glass substrate. With this orientation, it is believed that particles can be either driven into the slide, or pulled off of the slide, allowing us to create samples with precise particle densities. Due to the nature of the negatively charged particles, when the field is directed into the sample (the side which will be imaged), the particles are being driven away from the surface of the substrate, and vice versa for a field directed away from the sample.

3.1.1 Modeling Results

The CSAE model is defined on a two-dimensional lattice, with two possibilities for site occupation. A site can either be occupied, with corresponding occupation number, $n_i = 1$, or can be unoccupied, with corresponding occupation number, $n_1 = 0$. An example of the lattice can be seen in Figure 2.6.

The transition rate for this model is defined to be:

$$c(n_i \to (1 - n_i)) = (1 - n_i)(1 - \beta\frac{1}{z}\sum_{j\in NN} n_j) + \gamma_1(1 - n_i) + \gamma_2(n_i) \tag{3.1}$$

For this transition rate, β defines the interaction between the nearest neighbors of site i, where $0 \leq \beta \leq 1$. The higher β is, the more electrostatic repulsion exists between the particles. For $\beta = 0$, there is no interaction between the particles. $\sum n_j$ indicates the number of nearest neighbors of site i. For a two-dimensional lattice, the maximum number of nearest neighbors any particular site may have is four. In this transition rate, z is defined to be the total possible number of nearest neighbors (again, for a two-dimensional lattice, this is four).

Finally, γ_1 defines the probability of particle attachment to the slide due to an external field, while γ_2 defines the probability of particle detachment due to an external field. These can either be constants, or functions of time, allowing us to model a variety of electric fields. This model can be extended to an N-dimensional lattice, which would account for multiple bilayers, but for the purposes of this study, only monolayers were investigated.

3.1.2 Simulation Results

By employing the mean-field technique presented in [26], we are able to derive an equation for the total particle density through the ensemble average of $\langle n_i\rangle$, which allows us to approximate $\langle n_i n_j\rangle = \langle n_i\rangle\langle n_j\rangle$. This gives us:

$$\frac{\partial\langle n_i\rangle}{\partial t} = -\gamma_2\langle n_i\rangle + \gamma_1(1 - \langle n_i\rangle) + (1 - \langle n_i\rangle)(1 - \beta\frac{1}{z}\langle\eta\rangle) \tag{3.2}$$

where $\langle\eta\rangle = \sum_{j\in NN}\langle n_j\rangle$, which is the average of the sum of the nearest neighbors of site i.

We assume translational symmetry across the lattice, so we obtain the average site density, $\rho_i = \langle n_i\rangle$, which is equivalent to the total particle density $\rho = \sum_i n_i/N$, and N is equal to the total number of sites on the lattice.

Substituting this into equation (3.2), we are left with the time dependent particle density relation:

$$\frac{\partial\rho}{\partial t} = -\gamma_2\rho + \gamma_1(1 - \rho) + (1 - \rho)(1 - \beta\rho) \tag{3.3}$$

For the steady-state case, we find an overall particle density of:

$$\rho = \frac{\pm\sqrt{(\beta + \gamma_1 + \gamma_2 + 1)^2 - 4\beta(\gamma_1 + 1)} + \beta + \gamma_1 + \gamma_2 + 1}{2\beta} \tag{3.4}$$

with $\beta \neq 0$, and

$$\rho = \frac{\gamma_1 + 1}{\gamma_1 + \gamma_2 + 1} \tag{3.5}$$

for $\beta = 0$, and $\gamma_1 + \gamma_2 + 1 \neq 0$.

Using a graphical simulator such as Mathematica, Maple, or Python, we are able to obtain numerical solutions to the above equations for steady state $\rho(\beta)$ with $\gamma_1 = 0$ and $\gamma_2 \neq 0$. This simulates particles being encouraged to detach from the lattice.

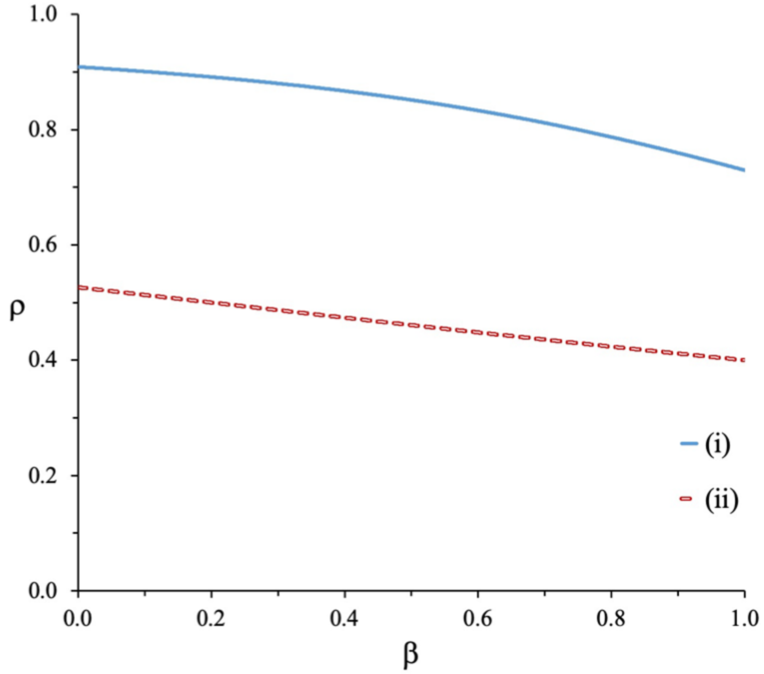


Figure 3.1: Results for steady-state particle density as a function of β, where $\gamma_1 = 0$, and two different values for γ_2. The blue line represents $\gamma_2 = 0.1$, while the dotted-red line represents $\gamma_2 = 0.9$. Reproduced from [15].

As can be seen from Figure 3.1, the steady-state particle density changes as a function of β, where β accounts for electrostatic repulsion between particles. As expected, as the interactions between particles increase, the particle density goes down. This figure shows two different values for γ_2, to show how the particle density is affected by the presense of an external field. The blue line corresponds to $\gamma_2 = 0.1$, while the red dotted line corresponds to $\gamma_2 = 0.9$. We expect that as γ_2 increases, the overall particle density on the slide will decrease, due to an increased effect of particle detachment due to external fields, which is exemplified by the data in Figure 3.1.

The other trend that is seen by looking at the data in Figure 3.1 is that β has a greater effect when γ_2 is low. When there is less detachment of particles due to an external field, the interactions between particles has more of an impact on the overall particle density, shown by the curved line. The line $\gamma_2 = 0.9$ on the other hand is linear. With an increase in particle detachment due to external sources, the relationship between particle density and electrostatic repulsion becomes linear.

Using the time dependent equation with a constant γ_1 or γ_2, we are able to solve for the particle density over time. For this example, we use $\gamma_1 \neq 0$ and $\gamma_2 = 0$, which represents particles being encouraged to attach to the lattice. Solving for $\rho(t)$, we obtain the following equation:

$$\rho(t) = \frac{(1+\gamma_1)(1-e^{t(\beta-(1+\gamma_1))})}{(1+\gamma_1)-\beta e^{t(\beta-(1+\gamma_1))}} \tag{3.6}$$

The results for several different values of γ_1 and β can be seen in Figure 3.4, below.

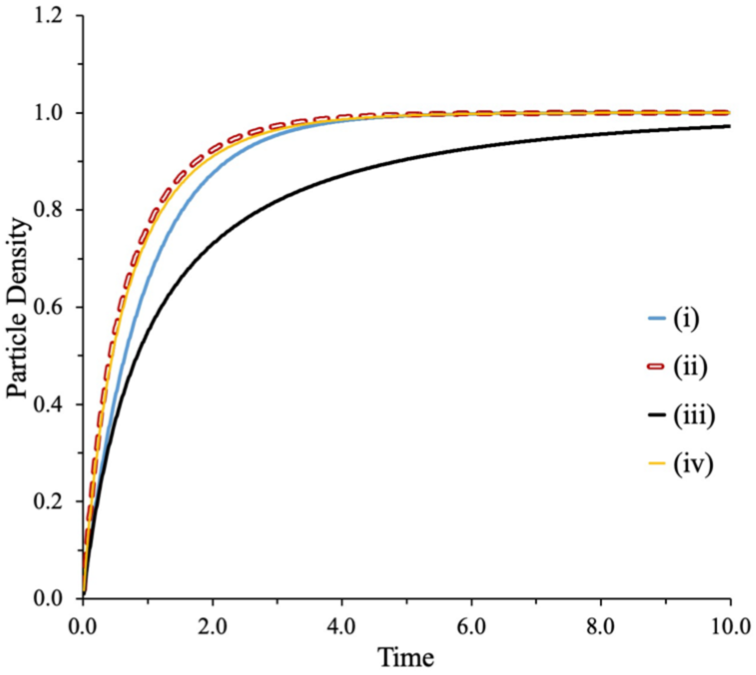


Figure 3.2: Results for particle density versus time for four different values for γ_1 and β: (i) $\gamma_1 = 0.1$ and $\beta = 0.1$; (ii) $\gamma_1 = 0.9$ and $\beta = 0.1$; (iii) $\gamma_1 = 0.1$ and $\beta = 0.9$; (iv) $\gamma_1 = 0.8$ and $\beta = 0.9$. Reproduced from [15].

Due to the particles being pushed into the slide for the above graph, the steady state partical density reaches 100% regardless of the parameters set, however the time to reach a 100% particle density is dependent upon those parameters. As can be seen above, full coverage is achieved most quickly with a strong external field, and is mostly independent of the electrostatic interaction between particles. Full coverage is achieved slowest for a weak external field and high electrostatic repulsion between particles, indicating that the factor that affects the time which it takes to achieve full coverage the most is the strength of the electric field.

Not only do these equations work for constant electric fields, but they can also be extrapolated to work for time-dependent external fields, such as an oscillatory electric field. This can be seen in Figure 3.3.

Due to the oscillatory nature of the field, the particle density of the lattice in this scenario never reaches 100%, and the saturation oscillates around a set value, which for this particular

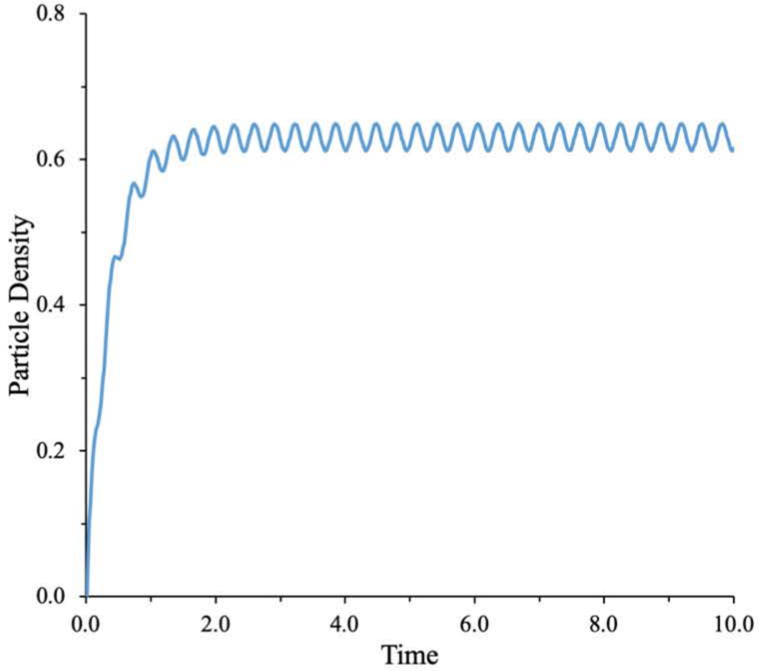


Figure 3.3: Results for the particle density over time under the presence of an oscillating field. The conditions used for this simulation are: $\gamma_1 = 0.5$; $\gamma_2 = 0.4$; $\beta = 0.5$; oscillating field is represented by $\gamma_2 \cos(10t)$. Reproduced from [15].

example is around 60%.

This method of determining particle density can be extrapolated to three-dimensional lattices, representing a lattice with multiple layers of nanoparticles.

3.1.3 Experimental Results

These models apply to thin-films created by layer-by-layer self-assembly under the influence of external fields, or can model the particle density of thin-films created with no external fields. The dipping process for creating thin-films under the influence of external fields is done in the same way as outlined in Figure 2.1, with the exception that during the deposition stage (the second cylindrical container), a field is placed across the slide. This can be done with a staining jar with copper tape on opposite sides of the jar, to create a parallel-plate capacitor. The deposition process under this condition is shown in Figure 3.4, and the jar may be seen in Figure 3.5.

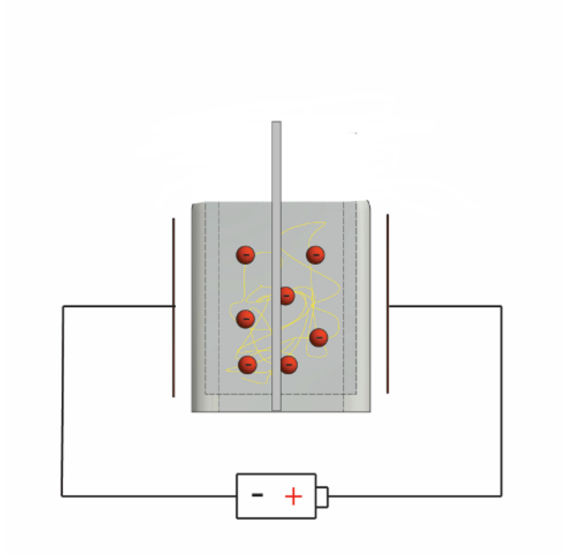


Figure 3.4: Parallel-plate capacitor across a staining jar, with field oriented perpendicular to the slide.

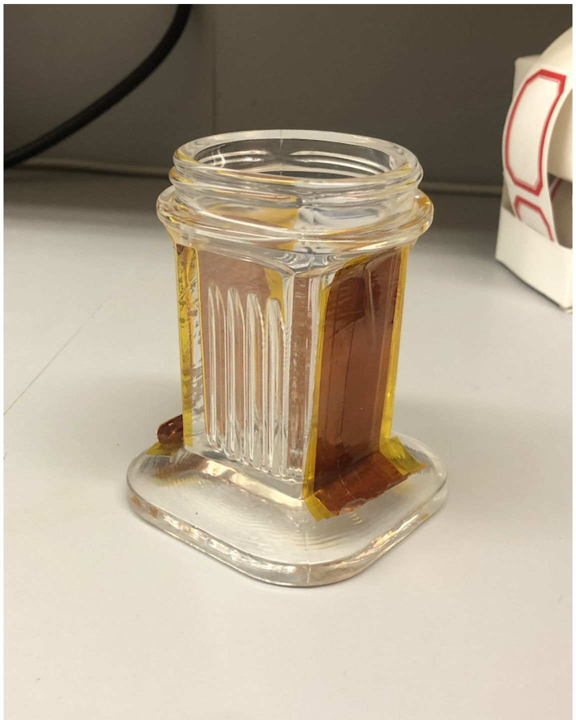


Figure 3.5: Parallel-plate capacitor made of copper tape across a staining jar, with field oriented perpendicular to the slide. The yellow tape is insulating tape to ensure safety due to the high voltages being applied across the capacitor

Using monolayer samples created in this manner, we are able to analyze them using SEM imagery. Using an imaging process through the computer program, ImageJ, we are able to obtain a particle density of sections of the sample. A table with our results may be seen in Table 3.1.

Sample	Field In Particle Density	Field Out Particle Density
1	47.87	44.46
2	40.41	30.58

Table 3.1: Average particle density in % for two samples produced under the presence of a 10,000 V potential difference across the capacitor. The field directed into the slide encourages particle detachment due to the negative nature of the silica nanoparticles, while a field directed away from the slide encourages particle attachment.

As can be seen from Table 3.1, the results of this exploratory experiment were generally inconclusive. Though we were able to apply a potential difference of 10,000 Volts, resulting in an electric field on the order of 254 kV/m across the parallel-plate capacitor, we found that the particle density is not substantially different despite the two different orientations of the electric field. Due to the lack of results, we ceased following this line of investigation.

3.2 Titania Nanoparticles

Although it is typical for thin-films to be created of bilayers made from silica nanoparticles, there are also instances of thin-films being created using titania (TiO_2) nanoparticles [29] [30], as well as using a mix of titania and silica nanoparticles [31]. The purpose of using different types of nanoparticles is to obtain thin-films with different optical properties, tailored to the needs of the specific experiment being performed.

3.2.1 Silica and Titania Mixes

Through exploratory experiments, we sought to make good quality thin films with either just titania nanoparticles or of mixes of silica and titania nanoparticles. We used two different types of titania in order to create these thin films - titania particles in a powder form, as well as a liquid suspension of titania nanoparticles. The liquid nanoparticles used are 5-30 nm rutile titanium oxide nanoparticles in a water dispersion from Nanostructured and Amorphous Materials Inc. (NanoAmor), and the powder nanoparticles used are Aeroxide TiO_2 P25 produced by Evonik Industries. Evonik Industries states that the specific surface area is 35-65 m^2/g [32], and we estimate that the diameter of the particles is around 35 nm.

Using both of these kinds of nanoparticles, the first thing we needed to determine is what concentration of nanoparticle to DI water produces the best quality thin films. Beginning with powder TiO_2, we began at a concentration of 100 milliliters of DI water to 0.05 grams of powder TiO_2. We found that the solution was rather opaque, whereas the silica 100:1 solution is quite transparent, indicating that the concentration of nanoparticles is too high. Diluting the solution by half each time, we obtained five samples of a single monolayer of titania nanoparticles ranging in concentration from 100:0.05 all the way to 800:0.05.

To determine the quality of the slides, we compared the transmission and reflection spectra to that of a single monolayer of 100:1 silica nanoparticles. The results of which may be seen in Figure 3.6.

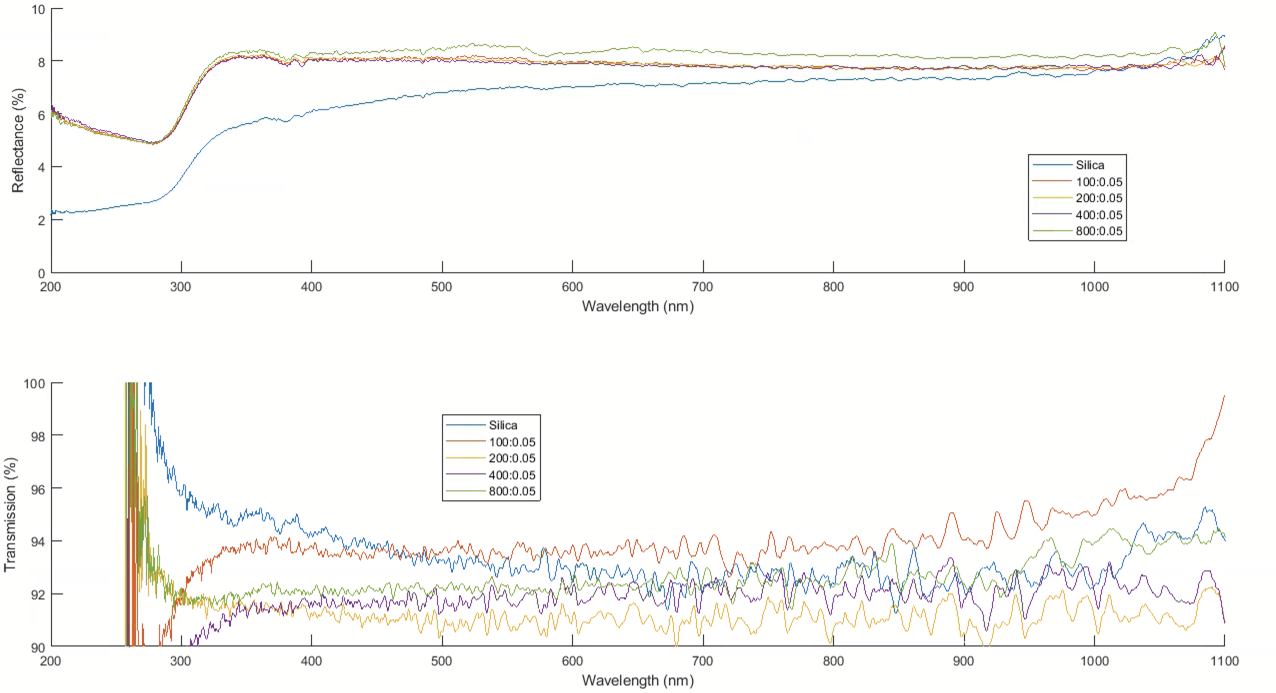


Figure 3.6: Reflectance (top) and transmittance (bottom) of several different concentrations of powder titania nanoparticles dispersed in deionized water as compared to a 100:1 concentration of silica nanoparticles.

From the reflectance spectra of the titania nanoparticles, it is evident that the reflectance is significantly higher than that of the silica, particularly at lower wavelengths. Additionally, the reflectance of a clean glass substrate is approximately 8%, which is about the reflectance seen from the powder titania samples.

This same line of inquiry was followed for the liquid nanoparticles as well. When the liquid nanoparticle solution was diluted to a concentration of 100 milliliters of deionized water to 1 milliliter of titania nanoparticles, the solution was again rather opaque. This is because the suspension of titania nanoparticles is more dense than that of the silica. The concetration was again diluted by half until we reached a concentration fo 800:1. The transmission and reflection at these different concentrations may be seen compared to that of the silica in Figure 3.7.

The reflectance specrtra of this data shows a similar trend to that of the powder titania nanoparticles, with the exception that the 100:1 sample generally follows the same trend as that of the silica nanoparticles. The same general trend can be seen in the transmission spectra, where the titania trends towards $\approx 92\%$, which is the transmission of a clean glass slide, and the silica is higher than that. That being said, in the transmission spectra of the 100:1 liquid titania solution, the transmission is higher than that of the silica, potentially indicating a better overall transmission of light, or indicating a higher amount of void space within the thin-film.

In order to better understand the optical properties of the titania nanoparticles, we generated

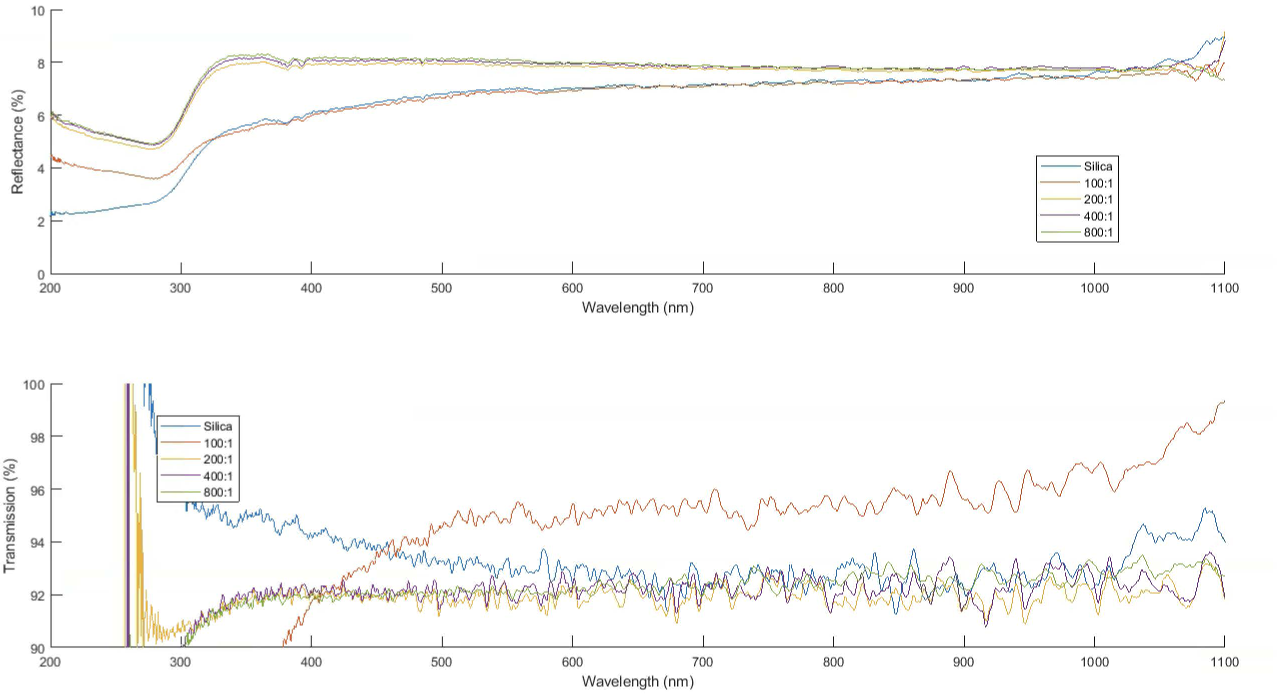


Figure 3.7: Reflectance (top) and transmittance (bottom) of several different concentrations of liquid titania nanoparticles dispersed in deionized water as compared to a 100:1 concentration of silica nanoparticles.

samples of 2, 4, 6, and 8 bilayers of the original concentrations of both the liquid and powder titania nanoparticle suspensions. We compared these results to that of the silica nanoparticles at each of the same number of bilayers for both reflectance and transmittance. These results may be seen in Figures 3.8 and 3.9 for liquid and powder, respectively.

As can be seen in Figure 3.8, the reflectance spectrum for both of the 2 bilayer samples are relatively similar, however the reflectance of the titania samples is higher for both 4, and 6 bilayers than the silica samples. The reflectance for the 8 bilayer sample of titania, however, is lower than that of the silica. For the transmission, all of the samples are around 95% at higher wavelengths, with the exception of the 8 bilayer silica sample which increases relatively linearly from 75% to 90%. The major difference in the transmission spectra between the silica samples and the titania samples is that the silica samples start at higher transmittances and tends towards 95% at higher wavelengths, whereas the titania begins at lower transmittances and tends towards 95% at higher wavelengths.

The powder samples, on the other hand, have a wider variation of trends across the wavelength spectrum for both reflectance and transmittance. The reflectances range from around 6% to 8% for the visible spectrum, but are lower at shorter wavelengths. The transmittances on the other hand have a much wider variation, ranging from ≈ 88% for 2 bilayers to ≈ 80% for 8 bilayers.

To get a better idea of the difference between the reflectance and the transmittance of the three different types of samples (silica, powder titania, and liquid titania), we graphed the reflectance added to the transmittance. In general, the two should add up to 100, so variations in this can

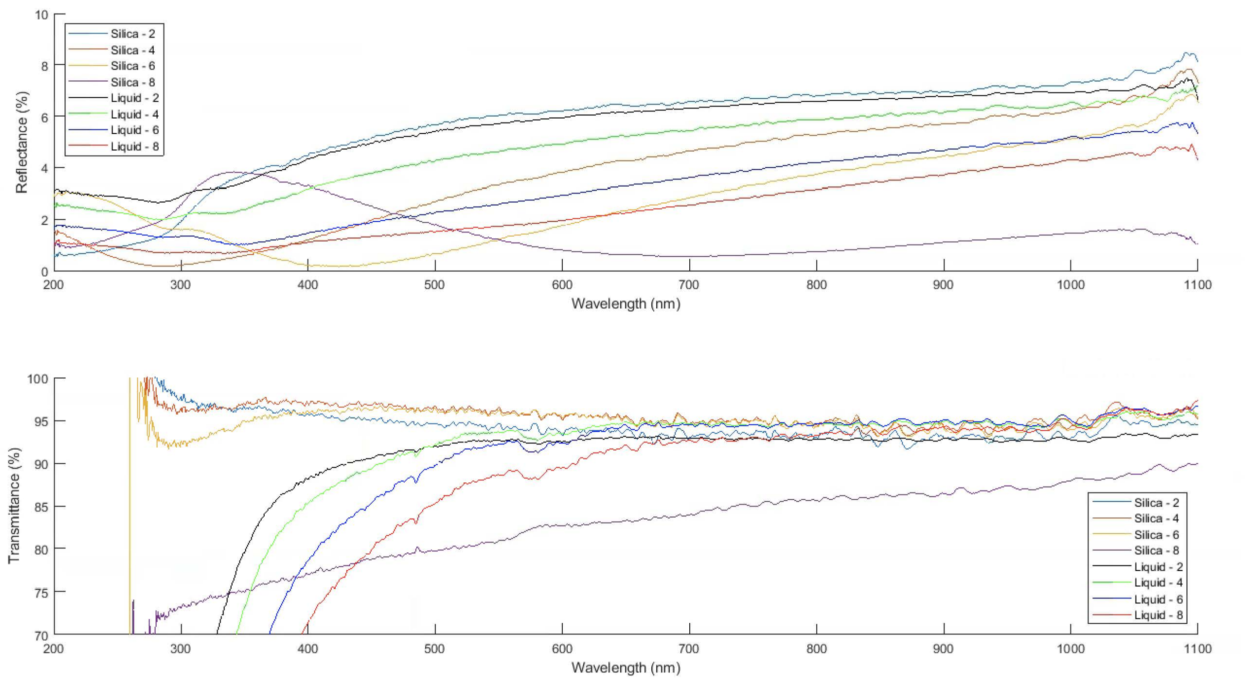


Figure 3.8: Reflectance (top) and transmittance (bottom) of 2, 4, 6, 8 bilayers of both silica and liquid titania nanoparticles.

indicate problems or losses within the system. This can be seen for all three types of samples in Figure 3.10.

From the top subplot in Figure 3.10, we see that (with the exception of the 8 bilayer sample), the $R + T$ for the silica bilayer samples is around 100%, as expected. The reason the 8 bilayer sample may be lower than the others could be because of diffuse reflection in the sample, or losses within the system, although it is hard to know for sure. For the two different types of titania on the other hand, the $R + T$ is quite a bit lower than 100%, particularly at shorter wavelengths.

We hypothesize that the reason for this is due to diffuse scattering (scattering of light in different directions due to an uneven surface). The spot size of the receiver on the thin-film analyzer is very small, so even small amounts of diffuse scattering would result in lower reflectances. The titania particles are larger than the silica nanoparticles, and have a tendency to clump together while on the slide, thus creating scattering. Another possible explanation for this is due to adsorption within the film.

3.2.2 Modeling Results

The data presented in Section 2.2 looks quite different than the data presented in Section 3.2.1. The data in this section is much more linear (at longer wavelengths), while the data in Section 2.2 is much more oscillatory in nature. The method presented in [21], which we used to determine the optical constants of the silica bilayer samples, relies on the fringes seen in Figure 2.3 and Figure 2.4, which are not seen in any of the figures presented in this section. The lack of fringes, or even a clear minima mean that we cannot extrapolate this method to work for these

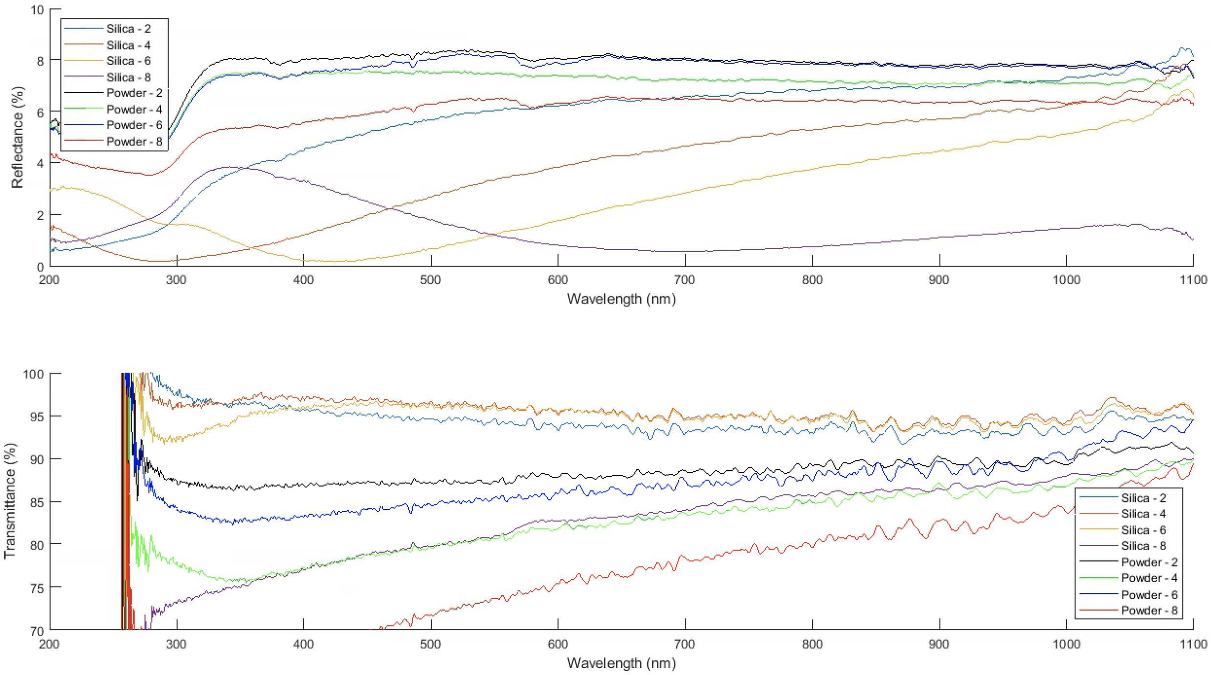


Figure 3.9: Reflectance (top) and transmittance (bottom) of 2, 4, 6, 8 bilayers of both silica and powder titania nanoparticles.

samples, and therefore we are unable to obtain an index of refraction.

3.2.3 Experimental Results

While creating samples for this section of this thesis, there were many observations we made. Primarily, that the suspensions of nanoparticles mixed with DI water had a tendency to clump together., despite being constantly stirred with a magnetic stirrer and a stir bar which sits inside the suspension. This indicates that the nanoparticles will likely clump together while deposited onto the slide as well.

In addition to this, the samples themselves, once dried, were cloudy in appearance. In general, the thin-films produced should be relatively clear, with a colorful tint when tilted in the light. These samples did not present these qualities, and instead looked like they degraded the quality of the image as opposed to making it better when they were looked through. This is thought to be because of the clumping and diffuse reflection as discussed at the end of section 3.2.1. A sample SEM image of TiO_2 on a glass substrate can be seen in Figure 3.11.

Due to the clumping of the titania nanoparticles, as well as the inability to produce a clear thin-film, this resulted in poor transmission and reflection data. The lack of fringes in this data meant we couldn't obtain values for the index of refraction, among other optical constants. For these reasons, this line of inquiry stopped, and we proceeded with silica bilayers.

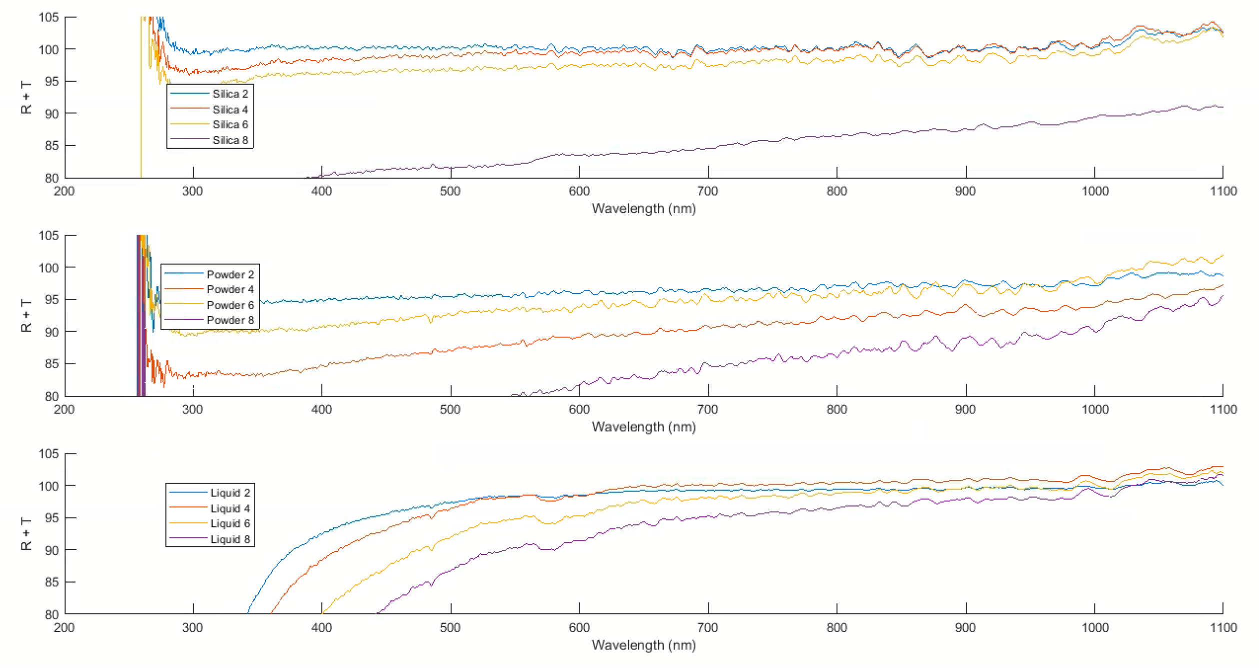


Figure 3.10: Reflectance plus transmittance for silica (top), powder titania nanoparticles (middle) and liquid titania nanoparticles (bottom).

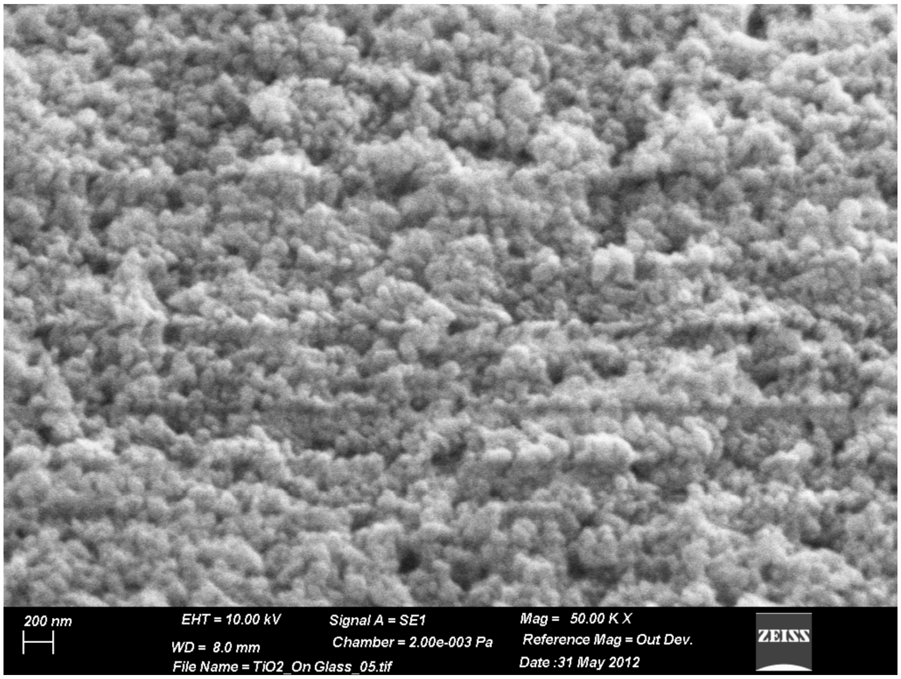


Figure 3.11: Sample SEM image of titania nanoparticles on a glass substrate. It should be noted that this sample is for an unknown number of bilayers, thought it is known that there is more than a single bilayer on this sample.

Chapter 4

Conclusion

In this thesis we present a class of models, simulation results, as well as experimental data for different types of self assembled thin-films of nanoparticles. Using ionic self-assembly of monolayers (ISAM), we create uniform layers of nanoparticles, which can be built up to create multilayer films that have different optical properties. Through the method presented in [21], we are able to calculate an important optical constant of our samples, namely the index of refraction, by looking at the transmission spectra of the sample as detailed in Chapter 2.

Through computer simulations, we are able to utilize cooperative sequential adsorption with evaporation (CSAE) models to predict the final particle density of our samples. The model is defined on a lattice with a transition rate that takes into account the possibility for evaporation of particles from the lattice. The model allows for the deposition of particles, taking into account the number of nearest neighbors at each site, as well as the interaction between particles within the layers and between layers. From this model, we are able to determine the particle density as a function of the number of bilayers which we compare to the index of refraction as a function of the number of bilayers in Section 2.5. In comparing the particle density computer simulations and the index of refraction determined experimentally, we determine that while the correlation between the two was not perfect, it describes the relationship as anticipated.

We performed two exploratory experiments through the course of this thesis - both of which gave inconclusive results. The first of the two experiments attempted to control the particle density of silica nanoparticles on a glass substrate through the use of electric fields. Through SEM imagery and image analysis, we determined that the two different orientations did not produce statistically significant differences between the particle densities.

The second exploratory experiment utilized titania nanoparticles in order to make thin-films with different optical properties than those of the silica nanoparticles. Through using the same technique that was used in the analysis of the silica bilayer slides, we were unable to calculate an index of refraction for these samples due to a lack of fringes. In addition, the samples looked very cloudy on the slide, as opposed to clear with a slight tint when tilted in the light, which is the desired outcome. We determined that the titania nanoparticles have a tendency to clump together, as opposed to dispersing uniformly across the slide, meaning that the films do not create an effective anti-reflective coating.

The results from the index of refraction for the silica nanoparticles have promising applications for the use of silica nanoparticles in the use of anti-reflective coatings. In the future, we can continue looking into these results in attempts to better control and refine the AR coatings. Additionally, we may look into utilizing magnetic fields in an attempt to control particle density on the slide, similarly to the exploratory experiments that were done with electric fields.

Bibliography

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

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

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

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

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

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

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

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

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

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

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

[12] 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

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

[14] 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

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

[16] Tomlin S G 1968 *J. Phys. D: Appl. Phys.* **1** 1667

[17] Manifacier J C, Gaslot J, Fillard J P 1976 *J. Phys. E: Sci. Instrum.* **9** 1002-1004

[18] Mahanty S, Roy S, Suchitra S 2003 *Journal of Crystal Growth* **261** 77-81

[19] Mahanty S, Basak D, Merino J M, Leon M 1999 *Materials Science and Engineering* **B68** 72-75

*References are hyperlinked in text!

[20] Hishikawa Y, Nakamura N, Tsuda A, Nakano S, Kishi Y, Kuwano Y 1991 *J. Appl. Phys.* **30** 1008

[21] Swanapoel R 1983 *J. Phys. E: Sci. Instrum.* **16** 1214

[22] Chaajed S, Poxson D J, Yan X, Cho J, Schubert E F, Welser R E, Sood A K, Kim J K 2001 *App. Phys. Ex.* **4** 052503

[23] Filmetrics 2008 *Thin-Film Analyzer F20: Operation Manual*

[24] Arfken G *Mathematical Methods for Physicists* (Academic Press)

[25] Hess M, Jones R G, Kahovec J, Kitayama T, Kratachvil P, Kubisa P, Mormann W, Stepto R F T, Tabak D, Vohlidal J, Wilks E S 2006 *Pure Appl. Chem.* **78** 2067-2074

[26] Mazilu D A, Mazilu I, Williams H T 2018 *From Complex to Simple* Morgan and Claypool Publishers

[27] Galy T, Marszewski M, King S, Yan Y, Tolbert S H, Pilon L 2019 *Microporous Mesoporous Mater.* **291** 109677

[28] Tamar Y, Tzabari M, Haspel C, Sasson Y 2014 *Sol. Energy Mater. Sol. Cells* **130** 246-256

[29] Wang Z-S, Sasaki T, Muramatsa M, Ebina Y, Tanaka T, Wang L, Watanabe M 2003 *Chem. Mater.* **15** 807-812

[30] Zhao L, Ming T, Li G, Chen H, Wang J, Yu J C 2010 *Small* **17** 1880-1885

[31] Kozhukharov S, Nenova Z, Nenov T, Nedev N, Machkova M 2015 *Sensors and Actuators B: Chem.* **210** 676-684

[32] Evonik Industries 2020 *Product Information* ***Aeroxide*** $\boldsymbol{TiO_2}$ ***P 25***

[33] Quote Investigator 2016

Appendix A

Sample Codes - MATLAB

This code allows us to import all of the data from the Filmetrics Thin-Film Analyzer into MATLAB, so that the data may be more easily plotted and manipulated, in order to calculate the index of refraction. It is important to note that the Thin-Film Analyzer outputs data into a ’.csv’ file, with the wavelength and the corresponding reflectance or transmittance. This data can be imported directly into MATLAB, but in this case, we merged all of the ’.csv’ files into one Excel spreadsheet for ease of importation, as well as initial plotting.

Going through the code, there are comments to help the viewer understand what is being done, but essentially the data is imported, and then $R + T$ is calculated (and normalized to 1, because the Thin-Film Analyzer has the data normalized to 100) so as to determine how close the value is to 1 (for any film, $R + T$ should be equal to 1, and if this is not the case, it indicates there may be extinction within the film). Then the data is plotted, and the maxima and minima of the transmittance spectra is found for the data sets which show fringes. Finally, the indices of refraction are found for each sample based on method outlined in Section 2.2.

```
%% Extract Reflectance Data from Excel Sheet T = importdata(’Thesis 2.xlsx’); % imports the excel sheet into Matlab
Wavelength = T.data.Reflectance(:,1); %extracts data from the correct column in the Excel sheet
R0 = T.data.Reflectance(:,3);
R1_1 = T.data.Reflectance(:,5);
R1_2 = T.data.Reflectance(:,6);
R1_3 = T.data.Reflectance(:,7);
R1_avg = T.data.Reflectance(:,7);
R2_1 = T.data.Reflectance(:,8);
R2_2 = T.data.Reflectance(:,9);
R2_3 = T.data.Reflectance(:,10);
R2_avg = T.data.Reflectance(:,11);
R3_1 = T.data.Reflectance(:,12);
R3_2 = T.data.Reflectance(:,13);
R3_3 = T.data.Reflectance(:,14);
R3_avg = T.data.Reflectance(:,15);
R4_1 = T.data.Reflectance(:,16);
R4_2 = T.data.Reflectance(:,17);
R4_3 = T.data.Reflectance(:,18);
R4_avg = T.data.Reflectance(:,19);
R5_1 = T.data.Reflectance(:,20);
R5_2 = T.data.Reflectance(:,21);
R5_3 = T.data.Reflectance(:,22);
R5_avg = T.data.Reflectance(:,23);
R6_1 = T.data.Reflectance(:,24);
R6_2 = T.data.Reflectance(:,25);
R6_3 = T.data.Reflectance(:,26);
R6_avg = T.data.Reflectance(:,27);
R7_1 = T.data.Reflectance(:,28);
R7_2 = T.data.Reflectance(:,29);
R7_3 = T.data.Reflectance(:,30);
R7_avg = T.data.Reflectance(:,31);
R8_1 = T.data.Reflectance(:,32);
R8_2 = T.data.Reflectance(:,33);
R8_3 = T.data.Reflectance(:,34);
R8_avg = T.data.Reflectance(:,35);
R9_1 = T.data.Reflectance(:,36);
R9_2 = T.data.Reflectance(:,37);
R9_3 = T.data.Reflectance(:,38);
R9_avg = T.data.Reflectance(:,39);
```

```
R10_1 = T.data.Reflectance(:,40);
R10_2 = T.data.Reflectance(:,41);
R10_3 = T.data.Reflectance(:,42);
R10_avg = T.data.Reflectance(:,43);
R11_1 = T.data.Reflectance(:,44);
R11_2 = T.data.Reflectance(:,45);
R11_3 = T.data.Reflectance(:,46);
R11_avg = T.data.Reflectance(:,47);
R12_1 = T.data.Reflectance(:,48);
R12_2 = T.data.Reflectance(:,49);
R12_3 = T.data.Reflectance(:,50);
R12_avg = T.data.Reflectance(:,51);
R13_1 = T.data.Reflectance(:,52);
R13_2 = T.data.Reflectance(:,53);
R13_avg = T.data.Reflectance(:,54);
R14_1 = T.data.Reflectance(:,55);
R14_2 = T.data.Reflectance(:,56);
R14_3 = T.data.Reflectance(:,57);
R14_avg = T.data.Reflectance(:,58);
R15_1 = T.data.Reflectance(:,59);
R15_2 = T.data.Reflectance(:,60);
R15_3 = T.data.Reflectance(:,61);
R15_avg = T.data.Reflectance(:,62);
R16_1 = T.data.Reflectance(:,63);
R16_2 = T.data.Reflectance(:,64);
R16_3 = T.data.Reflectance(:,65);
R16_avg = T.data.Reflectance(:,66);
%% Extract Transmittance Data from Excel Sheet
t0 = T.data.Transmittance(:,3); %same as above but for the transmittance
t1_1 = T.data.Transmittance(:,4);
t1_2 = T.data.Transmittance(:,5);
t1_3 = T.data.Transmittance(:,6);
t1_avg = T.data.Transmittance(:,7);
t2_1 = T.data.Transmittance(:,8);
t2_2 = T.data.Transmittance(:,9);
t2_3 = T.data.Transmittance(:,10);
t2_avg = T.data.Transmittance(:,11);
t3_1 = T.data.Transmittance(:,12);
t3_2 = T.data.Transmittance(:,13);
t3_3 = T.data.Transmittance(:,14);
t3_avg = T.data.Transmittance(:,15);
t4_1 = T.data.Transmittance(:,16);
t4_2 = T.data.Transmittance(:,17);
t4_3 = T.data.Transmittance(:,18);
t4_avg = T.data.Transmittance(:,19);
t5_1 = T.data.Transmittance(:,20);
t5_2 = T.data.Transmittance(:,21);
t5_3 = T.data.Transmittance(:,22);
t5_avg = T.data.Transmittance(:,23);
t6_1 = T.data.Transmittance(:,24);
t6_2 = T.data.Transmittance(:,25);
t6_3 = T.data.Transmittance(:,26);
t6_avg = T.data.Transmittance(:,27);
t7_1 = T.data.Transmittance(:,28);
t7_2 = T.data.Transmittance(:,29);
t7_3 = T.data.Transmittance(:,30);
t7_avg = T.data.Transmittance(:,31);
t8_1 = T.data.Transmittance(:,32);
t8_2 = T.data.Transmittance(:,33);
t8_3 = T.data.Transmittance(:,34);
t8_avg = T.data.Transmittance(:,35);
t9_1 = T.data.Transmittance(:,36);
t9_2 = T.data.Transmittance(:,37);
t9_3 = T.data.Transmittance(:,38);
t9_avg = T.data.Transmittance(:,39);
t10_1 = T.data.Transmittance(:,40);
t10_2 = T.data.Transmittance(:,41);
t10_3 = T.data.Transmittance(:,42);
t10_avg = T.data.Transmittance(:,43);
t11_1 = T.data.Transmittance(:,44);
t11_2 = T.data.Transmittance(:,45);
t11_3 = T.data.Transmittance(:,46);
t11_avg = T.data.Transmittance(:,47);
t12_1 = T.data.Transmittance(:,48);
t12_2 = T.data.Transmittance(:,49);
t12_3 = T.data.Transmittance(:,50);
t12_avg = T.data.Transmittance(:,51);
t13_1 = T.data.Transmittance(:,52);
t13_2 = T.data.Transmittance(:,53);
t13_3 = T.data.Transmittance(:,54);
t13_avg = T.data.Transmittance(:,55);
t14_1 = T.data.Transmittance(:,56);
t14_2 = T.data.Transmittance(:,57);
t14_3 = T.data.Transmittance(:,58);
t14_avg = T.data.Transmittance(:,59);
t15_1 = T.data.Transmittance(:,60);
t15_2 = T.data.Transmittance(:,61);
t15_3 = T.data.Transmittance(:,62);
t15_avg = T.data.Transmittance(:,63);
t16_1 = T.data.Transmittance(:,64);
t16_2 = T.data.Transmittance(:,65);
t16_3 = T.data.Transmittance(:,66);
t16_avg = T.data.Transmittance(:,67);
%% Figure out R+T, and normalize to 1 as opposed to 100
RT0 = R0+t0;
RT1_1 = R1_1+t1_1;
RT1_2 = R1_2+t1_2;
RT1_3 = R1_3+t1_3;
RT2_1 = R2_1+t2_1;
RT2_2 = R2_2+t2_2;
RT2_3 = R2_3+t2_3;
RT3_1 = R3_1+t3_1;
```

```
RT3_2 = R3_2+t3_2;
RT3_3 = R3_3+t3_3;
RT4_1 = R4_1+t4_1;
RT4_2 = R4_2+t4_2;
RT4_3 = R4_3+t4_3;
RT5_1 = R5_1+t5_1;
RT5_2 = R5_2+t5_2;
RT5_3 = R5_3+t5_3;
RT6_1 = R6_1+t6_1;
RT6_2 = R6_2+t6_2;
RT6_3 = R6_3+t6_3;
RT7_1 = R7_1+t7_1;
RT7_2 = R7_2+t7_2;
RT7_3 = R7_3+t7_3;
RT8_1 = R8_1+t8_1;
RT8_2 = R8_2+t8_2;
RT8_3 = R8_3+t8_3;
RT9_1 = R9_1+t9_1;
RT9_2 = R9_2+t9_2;
RT9_3 = R9_3+t9_3;
RT10_1 = R10_1+t10_1;
RT10_2 = R10_2+t10_2;
RT10_3 = R10_3+t10_3;
RT11_1 = R11_1+t11_1;
RT11_2 = R11_2+t11_2;
RT11_3 = R11_3+t11_3;
RT12_1 = R12_1+t12_1;
RT12_2 = R12_2+t12_2;
RT12_3 = R12_3+t12_3;
RT13_1 = R13_1+t13_1;
RT13_2 = R13_2+t13_2;
RT13_3 = R13_2+t13_3;
RT14_1 = R14_1+t14_1;
RT14_2 = R14_2+t14_2;
RT14_3 = R14_3+t14_3;
RT15_1 = R15_1+t15_1;
RT15_2 = R15_2+t15_2;
RT15_3 = R15_3+t15_3;
RT16_1 = R16_1+t16_1;
RT16_2 = R16_2+t16_2;
RT16_3 = R16_3+t16_3;
figure;
hold on
plot(Wavelength, RT1_1/100);
plot(Wavelength, RT1_2/100);
plot(Wavelength, RT1_3/100);
plot(Wavelength, RT2_1/100);
plot(Wavelength, RT2_2/100);
plot(Wavelength, RT2_3/100);
plot(Wavelength, RT3_1/100);
plot(Wavelength, RT3_2/100, ’b’);
plot(Wavelength, RT3_3/100, ’k’);
plot(Wavelength, RT4_1/100, ’r’);
plot(Wavelength, RT4_2/100, ’y’);
plot(Wavelength, RT4_3/100, ’m’);
plot(Wavelength, RT5_1/100, ’c’);
plot(Wavelength, RT5_2/100, ’g’);
plot(Wavelength, RT5_3/100, ’:b’);
plot(Wavelength, RT6_1/100, ’:k’);
plot(Wavelength, RT6_2/100, ’:r’);
plot(Wavelength, RT6_3/100, ’:y’);
plot(Wavelength, RT7_1/100, ’:m’);
plot(Wavelength, RT7_2/100, ’:c’);
plot(Wavelength, RT7_3/100, ’:g’);
plot(Wavelength, RT8_1/100, ’–b’);
plot(Wavelength, RT8_2/100, ’–k’);
plot(Wavelength, RT8_3/100, ’–r’);
plot(Wavelength, RT9_1/100, ’–y’);
plot(Wavelength, RT9_2/100, ’–m’);
plot(Wavelength, RT9_3/100, ’–c’);
plot(Wavelength, RT10_1/100, ’–g’);
plot(Wavelength, RT10_2/100, ’:’);
plot(Wavelength, RT10_3/100, ’:’);
plot(Wavelength, RT11_1/100, ’:’);
plot(Wavelength, RT11_2/100, ’:’);
plot(Wavelength, RT11_3/100, ’:’);
plot(Wavelength, RT12_1/100, ’:’);
plot(Wavelength, RT12_2/100, ’:’);
plot(Wavelength, RT12_3/100, ’–’);
plot(Wavelength, RT13_1/100, ’–’);
plot(Wavelength, RT13_2/100, ’–’);
plot(Wavelength, RT13_3/100, ’–’);
plot(Wavelength, RT14_1/100, ’–’);
plot(Wavelength, RT14_2/100, ’–’);
plot(Wavelength, RT14_3/100, ’–’);
plot(Wavelength, RT15_1/100, ’.-r’);
plot(Wavelength, RT15_2/100, ’.-b’);
plot(Wavelength, RT15_3/100, ’.-k’);
plot(Wavelength, RT16_1/100, ’.-m’);
plot(Wavelength, RT16_2/100, ’.-y’);
plot(Wavelength, RT16_3/100, ’.-c’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance + Transmittance’);
set(gcf, ’color’, ’w’);
xlim([400 700]);
ylim([0.7 1.05]);
hold off
%% Plot Reflectances
figure;
hold on
plot(Wavelength, R1_1);
plot(Wavelength, R1_2);
plot(Wavelength, R1_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’1 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R2_1);
plot(Wavelength, R2_2);
plot(Wavelength, R2_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’2 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R3_1);
plot(Wavelength, R3_2);
plot(Wavelength, R3_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’3 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R4_1);
plot(Wavelength, R4_2);
plot(Wavelength, R4_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’4 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R5_1);
plot(Wavelength, R5_2);
plot(Wavelength, R5_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’5 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R6_1);
plot(Wavelength, R6_2);
plot(Wavelength, R6_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’6 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R7_1);
plot(Wavelength, R7_2);
plot(Wavelength, R7_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’7 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R8_1);
plot(Wavelength, R8_2);
plot(Wavelength, R8_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’8 Bilayer’);
hold off
figure;
hold on
```

```
plot(Wavelength, R9_1);
plot(Wavelength, R9_2);
plot(Wavelength, R9_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’9 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R10_1);
plot(Wavelength, R10_2);
plot(Wavelength, R10_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’10 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R11_1);
plot(Wavelength, R11_2);
plot(Wavelength, R11_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’11 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R12_1);
plot(Wavelength, R12_2);
plot(Wavelength, R12_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’12 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R13_1);
plot(Wavelength, R13_2);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’);
title(’13 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R14_1);
plot(Wavelength, R14_2);
plot(Wavelength, R14_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’14 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R15_1);
plot(Wavelength, R15_2);
plot(Wavelength, R15_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’15 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, R16_1);
plot(Wavelength, R16_2);
plot(Wavelength, R16_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Reflectance (%)’);
xlim([200 1100]);
ylim([0 10]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’16 Bilayer’);
hold off
%% Calculate T based on R (T of the film itself, not the whole system)
```

```
T0 = 1 - (R0/100); %This is the transmittance of the clean glass slide
T1_1 = 1 - (R1_1/100);
T1_2 = 1 - (R1_2/100);
T1_3 = 1 - (R1_3/100);
T1_avg = 1 - (R1_avg/100);
T2_1 = 1 - (R2_1/100);
T2_2 = 1 - (R2_2/100);
T2_3 = 1 - (R2_3/100);
T2_avg = 1 - (R2_avg/100);
T3_1 = 1 - (R3_1/100);
T3_2 = 1 - (R3_2/100);
T3_3 = 1 - (R3_3/100);
T3_avg = 1 - (R3_avg/100);
T4_1 = 1 - (R4_1/100);
T4_2 = 1 - (R4_2/100);
T4_3 = 1 - (R4_3/100);
T4_avg = 1 - (R4_avg/100);
T5_1 = 1 - (R5_1/100);
T5_2 = 1 - (R5_2/100);
T5_3 = 1 - (R5_3/100);
T5_avg = 1 - (R5_avg/100);
T6_1 = 1 - (R6_1/100);
T6_2 = 1 - (R6_2/100);
T6_3 = 1 - (R6_3/100);
T6_avg = 1 - (R6_avg/100);
T7_1 = 1 - (R7_1/100);
T7_2 = 1 - (R7_2/100);
T7_3 = 1 - (R7_3/100);
T7_avg = 1 - (R7_avg/100);
T8_1 = 1 - (R8_1/100);
T8_2 = 1 - (R8_2/100);
T8_3 = 1 - (R8_3/100);
T8_avg = 1 - (R8_avg/100);
T9_1 = 1 - (R9_1/100);
T9_2 = 1 - (R9_2/100);
T9_3 = 1 - (R9_3/100);
T9_avg = 1 - (R9_avg/100);
T10_1 = 1 - (R10_1/100);
T10_2 = 1 - (R10_2/100);
T10_3 = 1 - (R10_3/100);
T10_avg = 1 - (R10_avg/100);
T11_1 = 1 - (R11_1/100);
T11_2 = 1 - (R11_2/100);
T11_3 = 1 - (R11_3/100);
T11_avg = 1 - (R11_avg/100); %This is the transmittance of the film itself, normalized to 1 as opposed to 100
T12_1 = 1 - (R12_1/100);
T12_2 = 1 - (R12_2/100);
T12_3 = 1 - (R12_3/100);
T12_avg = 1 - (R12_avg/100);
T13_1 = 1 - (R13_1/100);
T13_2 = 1 - (R13_2/100);
T13_3 = 1 - (R13_2/100);
T13_avg = 1 - (R13_avg/100);
T14_1 = 1 - (R14_1/100);
T14_2 = 1 - (R14_2/100);
T14_3 = 1 - (R14_3/100);
T14_avg = 1 - (R14_avg/100);
T15_1 = 1 - (R15_1/100);
T15_2 = 1 - (R15_2/100);
T15_3 = 1 - (R15_3/100);
T15_avg = 1 - (R15_avg/100);
T16_1 = 1 - (R16_1/100);
T16_2 = 1 - (R16_2/100);
T16_3 = 1 - (R16_3/100);
T16_avg = 1 - (R16_avg/100);
%% Plot Transmittance
figure;
hold on
plot(Wavelength, T1_1);
plot(Wavelength, T1_2);
plot(Wavelength, T1_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’1 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T2_1);
plot(Wavelength, T2_2);
plot(Wavelength, T2_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’2 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T3_1);
plot(Wavelength, T3_2);
plot(Wavelength, T3_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
```

```
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’3 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T4_1);
plot(Wavelength, T4_2);
plot(Wavelength, T4_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’4 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T5_1);
plot(Wavelength, T5_2);
plot(Wavelength, T5_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’5 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T6_1);
plot(Wavelength, T6_2);
plot(Wavelength, T6_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’6 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T7_1);
plot(Wavelength, T7_2);
plot(Wavelength, T7_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’7 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T8_1);
plot(Wavelength, T8_2);
plot(Wavelength, T8_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’8 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T9_1);
plot(Wavelength, T9_2);
plot(Wavelength, T9_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’9 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T10_1);
plot(Wavelength, T10_2);
plot(Wavelength, T10_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’10 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T11_1);
plot(Wavelength, T11_2);
plot(Wavelength, T11_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
```

```
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’11 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T12_1);
plot(Wavelength, T12_2);
plot(Wavelength, T12_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’12 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T13_1);
plot(Wavelength, T13_2);
plot(Wavelength, T13_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’13 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T14_1);
plot(Wavelength, T14_2);
plot(Wavelength, T14_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’14 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T15_1);
plot(Wavelength, T15_2);
plot(Wavelength, T15_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’15 Bilayer’);
hold off
figure;
hold on
plot(Wavelength, T16_1);
plot(Wavelength, T16_2);
plot(Wavelength, T16_3);
set(gcf, ’color’, ’w’);
xlabel(’Wavelength (nm)’);
ylabel(’Transmittance (%)’);
xlim([200 1100]);
ylim([.9 1]);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’);
title(’16 Bilayer’);
hold off
%% Finding n, and whatnot (Do for T11 to test)
ts = T0(268:738); %extracts the data for the visible spectrum of the clean glass slide
Ts = mean(ts); %finds the mean of the transmittance of the clean glass slide (calculated to be .9201)
s = 1/Ts + (1/(Ts^2) − 1)^{1/2}; % calculates the index of refraction of the slide (calculated to be 1.5126)
% Tm6 = .9688; %minimum transmittance based on the graph - couldnt really get great dat for 6 bilayers
% TM6 = .9985; %maximum transmittance based on the graph
Tm7_1 = .9657;
Tm7_2 = .9681;
Tm7_3 = .9663;
Tm7_avg = .9668;
% TM7_avg = .9932; %only need max for the weak absorption region calculation
Tm8_1 = .9618;
Tm8_2 = .9636;
Tm8_3 = .9594;
Tm8_avg = .9616;
% TM8 = .9945;
Tm9_1 = .9555;
Tm9_2 = .9591;
Tm9_3 = .9567;
Tm9_avg = .9574;
% TM9 = .994;
Tm10_1 = .9559;
Tm10_2 = .9579;
Tm10_3 = .9605;
Tm10_avg = .9585;
% TM10 = .9921;
```

Tm11_1 = .9733;
Tm11_2 = .9724;
Tm11_3 = .9724;
Tm11_avg = .9657;
% TM11 = .999;
Tm12_1 = .9686;
Tm12_2 = .968;
Tm12_3 = .967;
Tm12_avg = .968;
% TM12 = .9997;
Tm13_1 = .9638;
Tm13_2 = .9634;
Tm13_avg = .9637;
% TM13 = .9995;
Tm14_1 = .958;
Tm14_2 = .9569;
Tm14_3 = .9576;
Tm14_avg = .9576;
% TM14 = .9996;
Tm15_1 = .9405;
Tm15_2 = .9404;
Tm15_3 = .9405;
Tm15_avg = .9406;
% TM15 = .9924;
Tm16_1 = .9443;
Tm16_2 = .9462;
Tm16_3 = .9429;
Tm16_avg = .9436;
% TM16 = .9926;
% transparent region
% $M6 = ((2*s)/(Tm6)) - (s^2+1)/2$; % this is for transparent region
% $n6 = (M6 + (M6^2 - s^2)^{1/2})^{1/2}$; % transparent region
% disp(n6)
$M7\_1 = ((2*s)/(Tm7\_1)) - (s^2+1)/2$;
$M7\_2 = ((2*s)/(Tm7\_2)) - (s^2+1)/2$;
$M7\_3 = ((2*s)/(Tm7\_3)) - (s^2+1)/2$;
$M7\_avg = ((2*s)/(Tm7\_avg)) - (s^2+1)/2$; % this is for transparent region
$n7\_1 = (M7\_1 + (M7\_1^2 - s^2)^{1/2})^{1/2}$;
$n7\_2 = (M7\_2 + (M7\_2^2 - s^2)^{1/2})^{1/2}$;
$n7\_3 = (M7\_3 + (M7\_3^2 - s^2)^{1/2})^{1/2}$;
$n7\_avg = (M7\_avg + (M7\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region
% disp(n7)
$M8\_1 = ((2*s)/(Tm8\_1)) - (s^2+1)/2$;
$M8\_2 = ((2*s)/(Tm8\_2)) - (s^2+1)/2$;
$M8\_3 = ((2*s)/(Tm8\_3)) - (s^2+1)/2$;
$M8\_avg = ((2*s)/(Tm8\_avg)) - (s^2+1)/2$; % this is for transparent region
$n8\_1 = (M8\_1 + (M8\_1^2 - s^2)^{1/2})^{1/2}$;
$n8\_2 = (M8\_2 + (M8\_2^2 - s^2)^{1/2})^{1/2}$;
$n8\_3 = (M8\_3 + (M8\_3^2 - s^2)^{1/2})^{1/2}$;
$n8\_avg = (M8\_avg + (M8\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region
% disp(n8)
$M9\_1 = ((2*s)/(Tm9\_1)) - (s^2+1)/2$;
$M9\_2 = ((2*s)/(Tm9\_2)) - (s^2+1)/2$;
$M9\_3 = ((2*s)/(Tm9\_3)) - (s^2+1)/2$;
$M9\_avg = ((2*s)/(Tm9\_avg)) - (s^2+1)/2$; % this is for transparent region
$n9\_1 = (M9\_1 + (M9\_1^2 - s^2)^{1/2})^{1/2}$;
$n9\_2 = (M9\_2 + (M9\_2^2 - s^2)^{1/2})^{1/2}$;
$n9\_3 = (M9\_3 + (M9\_3^2 - s^2)^{1/2})^{1/2}$;
$n9\_avg = (M9\_avg + (M9\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region
% disp(n9)
$M10\_1 = ((2*s)/(Tm10\_1)) - (s^2+1)/2$;
$M10\_2 = ((2*s)/(Tm10\_2)) - (s^2+1)/2$;
$M10\_3 = ((2*s)/(Tm10\_3)) - (s^2+1)/2$;
$M10\_avg = ((2*s)/(Tm10\_avg)) - (s^2+1)/2$; % this is for transparent region
$n10\_1 = (M10\_1 + (M10\_1^2 - s^2)^{1/2})^{1/2}$;
$n10\_2 = (M10\_2 + (M10\_2^2 - s^2)^{1/2})^{1/2}$;
$n10\_3 = (M10\_3 + (M10\_3^2 - s^2)^{1/2})^{1/2}$;
$n10\_avg = (M10\_avg + (M10\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region
% disp(n10)
$M11\_1 = ((2*s)/(Tm11\_1)) - (s^2+1)/2$;
$M11\_2 = ((2*s)/(Tm11\_2)) - (s^2+1)/2$;
$M11\_3 = ((2*s)/(Tm11\_3)) - (s^2+1)/2$;
$M11\_avg = ((2*s)/(Tm11\_avg)) - (s^2+1)/2$; % this is for transparent region
$n11\_1 = (M11\_1 + (M11\_1^2 - s^2)^{1/2})^{1/2}$;
$n11\_2 = (M11\_2 + (M11\_2^2 - s^2)^{1/2})^{1/2}$;
$n11\_3 = (M11\_3 + (M11\_3^2 - s^2)^{1/2})^{1/2}$;
$n11\_avg = (M11\_avg + (M11\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region
% disp(n11)
$M12\_1 = ((2*s)/(Tm12\_1)) - (s^2+1)/2$;
$M12\_2 = ((2*s)/(Tm12\_2)) - (s^2+1)/2$;
$M12\_3 = ((2*s)/(Tm12\_3)) - (s^2+1)/2$;
$M12\_avg = ((2*s)/(Tm12\_avg)) - (s^2+1)/2$; % this is for transparent region
$n12\_1 = (M12\_1 + (M12\_1^2 - s^2)^{1/2})^{1/2}$;
$n12\_2 = (M12\_2 + (M12\_2^2 - s^2)^{1/2})^{1/2}$;
$n12\_3 = (M12\_3 + (M12\_3^2 - s^2)^{1/2})^{1/2}$;
$n12\_avg = (M12\_avg + (M12\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region
% disp(n12)
$M13\_1 = ((2*s)/(Tm13\_1)) - (s^2+1)/2$;
$M13\_2 = ((2*s)/(Tm13\_2)) - (s^2+1)/2$;

$M13\_avg = ((2 * s)/(Tm13\_avg)) - (s^2 + 1)/2$; % this is for transparent region

$n13\_1 = (M13\_1 + (M13\_1^2 - s^2)^{1/2})^{1/2}$;

$n13\_2 = (M13\_2 + (M13\_2^2 - s^2)^{1/2})^{1/2}$;

$n13\_avg = (M13\_avg + (M13\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region

% disp(n13)

$M14\_1 = ((2 * s)/(Tm14\_1)) - (s^2 + 1)/2$;

$M14\_2 = ((2 * s)/(Tm14\_2)) - (s^2 + 1)/2$;

$M14\_3 = ((2 * s)/(Tm14\_3)) - (s^2 + 1)/2$;

$M14\_avg = ((2 * s)/(Tm14\_avg)) - (s^2 + 1)/2$; % this is for transparent region

$n14\_1 = (M14\_1 + (M14\_1^2 - s^2)^{1/2})^{1/2}$;

$n14\_2 = (M14\_2 + (M14\_2^2 - s^2)^{1/2})^{1/2}$;

$n14\_3 = (M14\_3 + (M14\_3^2 - s^2)^{1/2})^{1/2}$;

$n14\_avg = (M14\_avg + (M14\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region

% disp(n14)

$M15\_1 = ((2 * s)/(Tm15\_1)) - (s^2 + 1)/2$;

$M15\_2 = ((2 * s)/(Tm15\_2)) - (s^2 + 1)/2$;

$M15\_3 = ((2 * s)/(Tm15\_3)) - (s^2 + 1)/2$;

$M15\_avg = ((2 * s)/(Tm15\_avg)) - (s^2 + 1)/2$; % this is for transparent region

$n15\_1 = (M15\_1 + (M15\_1^2 - s^2)^{1/2})^{1/2}$;

$n15\_2 = (M15\_2 + (M15\_2^2 - s^2)^{1/2})^{1/2}$;

$n15\_3 = (M15\_3 + (M15\_3^2 - s^2)^{1/2})^{1/2}$;

$n15\_avg = (M15\_avg + (M15\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region

% disp(n15)

$M16\_1 = ((2 * s)/(Tm16\_1)) - (s^2 + 1)/2$;

$M16\_2 = ((2 * s)/(Tm16\_2)) - (s^2 + 1)/2$;

$M16\_3 = ((2 * s)/(Tm16\_3)) - (s^2 + 1)/2$;

$M16\_avg = ((2 * s)/(Tm16\_avg)) - (s^2 + 1)/2$; % this is for transparent region

$n16\_1 = (M16\_1 + (M16\_1^2 - s^2)^{1/2})^{1/2}$;

$n16\_2 = (M16\_2 + (M16\_2^2 - s^2)^{1/2})^{1/2}$;

$n16\_3 = (M16\_3 + (M16\_3^2 - s^2)^{1/2})^{1/2}$;

$n16\_avg = (M16\_avg + (M16\_avg^2 - s^2)^{1/2})^{1/2}$; % transparent region

% disp(n16)

% lambda1 = 287.7; % wavelength of one maximum

% lambda2 = 463.1; % wavelength of adjacent maximum

% d = (lambda1*lambda2)/(2*(lambda2*n-lambda1*n)); %calculate the thickness (d) of the slide

%disp(d)

N7 = table(n7_1, n7_2, n7_3, n7_avg);

N8 = table(n8_1, n8_2, n8_3, n8_avg);

N9 = table(n9_1, n9_2, n9_3, n9_avg);

N10 = table(n10_1, n10_2, n10_3, n10_avg);

N11 = table(n11_1, n11_2, n11_3, n11_avg);

N12 = table(n12_1, n12_2, n12_3, n12_avg);

N13 = table(n13_1, n13_2, n13_avg);

N14 = table(n14_1, n14_2, n14_3, n14_avg);

N15 = table(n15_1, n15_2, n15_3, n15_avg);

N16 = table(n16_1, n16_2, n16_3, n16_avg);

disp(N7)

disp(N8)

disp(N9)

disp(N10)

disp(N11)

disp(N12)

disp(N13)

disp(N14)

disp(N15)

disp(N16)

% S7 = std(N7);

% disp(S7)

%% Get the real part of n and plot versus # layers

N7_1 = real(n7_1);

N7_2 = real(n7_2);

N7_3 = real(n7_3);

N7_avg = real(n7_avg);

N8_1 = real(n8_1);

N8_2 = real(n8_2);

N8_3 = real(n8_3);

N8_avg = real(n8_avg);

N9_1 = real(n9_1);

N9_2 = real(n9_2);

N9_3 = real(n9_3);

N9_avg = real(n9_avg);

N10_1 = real(n10_1);

N10_2 = real(n10_2);

N10_3 = real(n10_3);

N10_avg = real(n10_avg);

N11_1 = real(n11_1);

N11_2 = real(n11_2);

N11_3 = real(n11_3);

N11_avg = real(n11_avg);

N12_1 = real(n12_1);

N12_2 = real(n12_2);

N12_3 = real(n12_3);

N12_avg = real(n12_avg);

N13_1 = real(n13_1);

N13_2 = real(n13_2);

N13_avg = real(n13_avg);

N14_1 = real(n14_1);

N14_2 = real(n14_2);

N14_3 = real(n14_3);

N14_avg = real(n14_avg);

N15_1 = real(n15_1);

N15_2 = real(n15_2);

N15_3 = real(n15_3);

```
N15_avg = real(n15_avg);
N16_1 = real(n16_1);
N16_2 = real(n16_2);
N16_3 = real(n16_3);
N16_avg = real(n16_avg);
x7 = 7;
x8 = 8;
x9 = 9;
x10 = 10;
x11 = 11;
x12 = 12;
x13 = 13;
x14 = 14;
x15 = 15;
x16 = 16;
figure;
hold on
scatter(x7, N7_1, ’r’);
scatter(x7, N7_2, ’b’);
scatter(x7, N7_3, ’g’);
scatter(x7, N7_avg, ’k’);
scatter(x8, N8_1, ’r’);
scatter(x8, N8_2, ’b’);
scatter(x8, N8_3, ’g’);
scatter(x8, N8_avg, ’k’);
scatter(x9, N9_1, ’r’);
scatter(x9, N9_2, ’b’);
scatter(x9, N9_3, ’g’);
scatter(x9, N9_avg, ’k’);
scatter(x10, N10_1, ’r’);
scatter(x10, N10_2, ’b’);
scatter(x10, N10_3, ’g’);
scatter(x10, N10_avg, ’k’);
scatter(x11, N11_1, ’r’);
scatter(x11, N11_2, ’b’);
scatter(x11, N11_3, ’g’);
scatter(x11, N11_avg, ’k’);
scatter(x12, N12_1, ’r’);
scatter(x12, N12_2, ’b’);
scatter(x12, N12_3, ’g’);
scatter(x12, N12_avg, ’k’);
scatter(x13, N13_1, ’r’);
scatter(x13, N13_2, ’b’);
scatter(x13, N13_avg, ’k’);
scatter(x14, N14_1, ’r’);
scatter(x14, N14_2, ’b’);
scatter(x14, N14_3, ’g’);
scatter(x14, N14_avg, ’k’);
scatter(x15, N15_1, ’r’);
scatter(x15, N15_2, ’b’);
scatter(x15, N15_3, ’g’);
scatter(x15, N15_avg, ’k’);
scatter(x16, N16_1, ’r’);
scatter(x16, N16_2, ’b’);
scatter(x16, N16_3, ’g’);
scatter(x16, N16_avg, ’k’);
xlabel(’Number Bilayers’);
ylabel(’Index of Refraction’);
legend(’Sample 1’, ’Sample 2’, ’Sample 3’, ’Average’, ’location’, ’best’);
xlim([6 17]);
set(gcf, ’color’, ’w’);
%% Plot the particle density versus number of layers
figure;
hold on
scatter(2, 60.9125, ’b’);
scatter(3, 72.1308, ’b’);
scatter(4, 77.1499, ’b’);
scatter(5, 80.1235, ’b’);
scatter(6, 82.0294, ’b’);
scatter(7, 83.2939, ’b’);
scatter(8, 84.6003, ’b’);
scatter(9, 85.3785, ’b’);
scatter(10, 85.7965, ’b’);
scatter(11, 85.9875, ’b’);
scatter(12, 86.3592, ’b’);
scatter(13, 86.6602, ’b’);
scatter(14, 86.8777, ’b’);
scatter(15, 87.0644, ’b’);
scatter(16, 87.2356, ’b’);
scatter(17, 87.4905, ’b’);
scatter(18, 87.6688, ’b’);
scatter(19, 87.9009, ’b’);
scatter(20, 88.0465, ’b’);
xlabel(’Number of Layers’);
ylabel(’Particle Density’);
xlim([1 20]);
set(gcf, ’color’, ’w’);
hold off
```

Appendix B

Sample Codes - Python

This code creates a lattice of a set size, which can be determined by the user, with a set number of layers. The evaporation rate, as well as the concentration of the particle bath, and the electrostatic screening coefficients can be set. Sites are updated on the lattice to depict particle interactions, and particle deposition on the slides dependent upon the outlined conditions. A lattice will be output which depicts the total number of particles in each site on the lattice (the maximum number of particles on a particular site is the number of layers of the film). In addition, a plot of the particle density over time will be output to indicate what the steady-state particle density is, and how long it takes (in arbitrary units) it takes to get to that density.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
”””
Created on Thu Jan 28 08:16:00 2021
@author: mazilui
”””
# -*- coding: utf-8 -*-
”””
Created on Wed Jan 27 20:36:30 2021
@author: Laurențiu Stoleriu
”””
import random
import numpy as np
import matplotlib.pyplot as plt
### system parameters
n_layers = 2 # total number of layers
dim_layer = 10 # one layer consists of (dim_layer x dim_layer) cells
### simulation parameters
tSteps = 1000 * (dim_layer * dim_layer * n_layers)
### physical parameters
gamma = 0.1 # evaporation rate
alpha1 = 0.1 # reservoir concentration
```

```
alpha2 = 0.1 # bulk concentration ################################# SHOULD
CHANGE WITH TOTAL OCCUPATION?
beta1 = 0.5 # electrostatic screening coefficients - within the layer
beta2 = 0.5 # electrostatic screening coefficients - between layers
### the system
layer = np.zeros((dim_layer, dim_layer, n_layers))
eta1 = np.zeros((dim_layer, dim_layer, n_layers)) # neighbors - within the layer #########
SHOULD BE NORMALIZED (EDGE EFFECS)?
eta2 = np.zeros((dim_layer, dim_layer, n_layers)) # neighbors - between layers
total = np.zeros(tSteps) # number of particles in the layer at each time step
random.seed(2021) # comment for randomness
def update_neighbours(i, j, k, val):
if (i == 0): #declare i-sites to update
list_i = [i, i+1]
elif (i == dim_layer-1):
list_i = [i-1, i]
else:
list_i = [i-1, i, i+1]
if (j == 0): #declare j-sites to update
list_j = [j, j+1]
elif (j == dim_layer-1):
list_j = [j-1, j]
else:
list_j = [j-1, j, j+1]
if (k == 0): #declare k-sites to update
list_k = [k+1]
elif (k == n_layers-1):
list_k = [k-1]
else:
list_k = [k-1, k+1]
for ii in list_i: # now that we have list of all sites, update no of neighbors
for jj in list_j:
eta1[ii, jj, k] = eta1[ii, jj, k] + val # decreasing no. of neighbors within the layer
for kk in list_k:
eta2[ii, jj, kk] = eta1[ii, jj, kk] + val # decreasing no. of neighbors between layers
contor_attached = 0
contor_evaporated = 0
for tStep in range(tSteps):
i = random.randint(0, dim_layer-1) # testing a random position [i, j]
j = random.randint(0, dim_layer-1)
k = 0 # starting with the lowest layer
while (k ¡ n_layers):
if (layer[i, j, k]): # if it's occupied go up one position and re-test
```

```
k = k+1
else:
break
### test evaporation
if (k == n_layers): # then this vertical stack is full
k = n_layers-1 # so, we’re testing evaporation on the n_layers-1 position
W = gamma
if (W ¿ random.random()):
layer[i, j, k] = 0 # it’s gone!
contor_evaporated = contor_evaporated + 1
update_neighbours(i, j, k, (-1))
total[tStep] = contor_attached - contor_evaporated
continue # move along, nothing else to see here #### OR WE CAN TEST RE-ATTACH HERE
### test attach
W = alpha1 * beta1 ** eta1[i, j, k] + alpha2 * beta2 ** eta2[i, j, k]
if (W ¿ random.random()):
layer[i, j, k] = 1 # welcome!
contor_attached = contor_attached + 1
update_neighbours(i, j, k, (+1))
total[tStep] = contor_attached - contor_evaporated
if( (tStep%1000) == 0 ):
print(f’Time step: {tStep} - attach: {contor_attached} - evap: {contor_evaporated}’)
if True: #swich off animation with ”if False:”
fig, ax = plt.subplots()
topview_matrix = np.zeros((dim_layer, dim_layer))
for i in range(dim_layer):
for j in range(dim_layer):
site_sum = 0
for k in range(n_layers):
site_sum = site_sum + layer[i, j, k]
topview_matrix[i, j] = site_sum
cax = ax.matshow(topview_matrix, vmin=0, vmax=n_layers, cmap=plt.cm.Blues, interpolation=’nearest’)
fig.colorbar(cax)
# adds ocuppation numbers in each cell if needed
for i in range(dim_layer):
for j in range(dim_layer):
c = topview_matrix[j,i]
ax.text(i, j, f’{c:.0f}’, va=’center’, ha=’center’)
# calculates the steady-state particle density based on the second half of the particle density data
middle_index = len(total)//2
halftotal = total[middle_index:]
avg = sum(halftotal)/len(halftotal)
```

```
print(avg/n_layers)
plt.figure()
plt.plot(total/n_layers)
plt.show()
```