# Biomedical Implant Corrosion Passivation Using PAMAM Dendrimer Films

## Final Report

Team Captain: Ben Jones
Design and Modeling: Elizabeth Ashley
Design and Modeling: Kerry Toole
Secretary and Prototyping: Rachel Stein
Treasurer and Editor: Mari Hagemeyer

**Abstract**

Common biomedical implants use biocompatible titanium alloys; however, research has shown that electrochemical corrosion of titanium results in titanium ion diffusion. These metal ions cause inflammation in surrounding tissue, prolonging convalescence and causing physical discomfort. Passivity of the devices has been shown to improve by adjusting alloy composition, oxide layer structure, and depositing polymer coatings. Dendrimer polymer films have been studied as antimicrobial coatings, but also have functional potential for use as corrosion passivation layers. Dendrimers are known to impede ion penetration into metal surfaces in solution, retarding the corrosion rate. A dendrimer film will be designed as a passivating layer for titanium biomedical implants. It will be fabricated on a biocompatible titanium substrate covered in a titanium oxide layer to passivate the surface and prevent diffusion of titanium ions into surrounding solution and to prevent diffusion of salt ions into the substrate.

**Table of Contents**

## Motivation

Titanium and its alloys are commonly used for biomedical implants because of their high strength and favorable biocompatibility. Because implants typically remain in the body for many years in a saline environment, the material used needs to be resistant to corrosion and degradation in physiological conditions. Titanium naturally forms a native oxide in any oxygen containing environment, including aqueous solutions that passivate the more reactive titanium underneath [Chaturvedi, 2009]. However, in solutions with chloride anions, the oxide coating can undergo pitting corrosion due to fluctuations in electrochemical potential across the oxide surface. This pitting corrosion can expose the titanium metal to the electrolytic solution, enabling the onset of galvanic corrosion [Pohler, 2000]. By adding a secondary passivation layer to the titanium oxide, corrosion of the implant and dissolution of titanium ions from the implant into the human body can be mitigated.

Dendrimers are fractal molecules that form globular shapes and internal cavities based upon electrostatic and steric repulsion between the branches. Poly(amidoamine) (PAMAM) dendrimers have been considered for drug delivery and antimicrobial purposes because of their relative biocompatibility [Wang et al., 2013]. Additionally, PAMAM's steric hindrance decreases the diffusion of ions through monolayers of the dendrimer. By coating implants that do not undergo wear with PAMAM, an additional passivating layer may be added to reduce the pitting corrosion of titanium oxide and extend the lifetime of biomedical implants.

## Aspects of Materials Science and Engineering

The design of this project, which can be subdivided into modeling and prototyping, relies heavily on principles of materials science and engineering. Considerations include corrosion and diffusion properties of biomedical metals (specifically titanium), the material properties of the oxide layer formed on the titanium, and fabrication of the dendrimer layer upon the oxide surface. The design must take into account potential chemical reactions that can occur between physiological chemicals and implant materials and how these reactions can result in corrosion. The design also takes under consideration the energies and hopping rates involved when ions diffuse into a dendrimer layer and through a dendrimer layer onto a titanium oxide surface. Effective simulation of ion diffusion requires working knowledge of material kinetics and thermodynamics.

Device prototyping relies upon a variety of microprocessing techniques, including sputtering, plasma oxidation, and thermal oxidation. Prototype characterization is accomplished via a wide range of materials science based techniques including

ellipsometry, scanning electron microscopy (SEM), and electron-dispersive x-ray spectroscopy (EDS).

## Previous Work

### Titanium Alloys for Biomedical Implants

Titanium is a commonly used metal in orthopedic and dental implants [Casaletto et al., 2001]. Its favorable biocompatibility, high strength and beneficial corrosion resistance makes titanium and its alloys more suitable for biomedical applications than other compatible metals such as stainless steel and Co-Cr alloys [Niinomi, 2008]. The most common uses for titanium include hard tissue replacement and reinforcement where high strength and long fatigue life is required [Niinomi, 2008]. Due to its mechanical strength and low density, titanium has an advantage over polymers for weight-bearing applications. Additionally, titanium can be alloyed without using nickel to prevent the possibility of allergic reactions [Pohler, 2000]. However, the long-term stability of titanium implants depends on the resistance of the implant material to corrosion, inhibition of bacterial adhesion to the implant, and integration of the artificial material with surrounding bone and tissue [Casaletto et al., 2001]. In our project, we are concerned with the degradation of titanium implants in a physiological solution for cases where the implant does not undergo bone integration or long term physical wear, such as in bone plates and screw surfaces.

There is a wide range of pre-existing research regarding the corrosive behavior and surface passivation mechanisms for metallic alloys used for biomedical implants [Josephs et al., 2009; Lausmaa et al., 1990; Pan et al., 1998; Schmutz, 2008]. Oxide coatings on metal implants are used to create a passive surface on the anodic metal, which prevents corrosion and degradation of the implant [Chaturvedi, 2009]. While oxide passivation layers are often used to decrease ion dissolution in biomedical implants, these layers are not infallible and are prone to pitting corrosion when immersed in bodily fluids over time [Chaturvedi, 2009]. Studies of the corrosive behavior of biomedical alloys over time have been conducted in a variety of simulated bodily fluids. In a study conducted by Vidal et al., the group determined that the oxide coatings on the alloys had undergone local changes in composition. [Vidal et al., 2008]. This change in composition was dependent both on the solution composition and on the type of alloy, even though all alloys were prone to pitting corrosion and metal ion dissolution. Due to the tendency of metals to corrode in aqueous environments, additional surface modifications to titanium and other metal implants were made to improve material biocompatibility and the product lifetime [Singh and Dahotre, 2007].

Titanium spontaneously forms a thin-film oxide of approximately 2 nm thickness on its surface whenever oxygen is present, creating a barrier between the highly reactive titanium metal and the surrounding environment [Basame and White, 1999]. This titanium oxide film protects the titanium surface from corrosion by limiting exposure of the anodic titanium metal to the environment, thus decreasing metal ion diffusion into the surroundings [Pohler, 2000]. In a physiological environment, the corrosion process can be accelerated due to the high concentration of chlorine ions and amino acids in biological fluids [Hanawa, 1999]. Simple oxidation of the titanium surface that is exposed to biological fluids decreases the corrosion rate and the diffusion of ions between the solution and the metal surface [Pan et al., 1998]. While titanium ion dissolution into the surrounding environment through the oxide coating is very small, it becomes significant for cases with large surface area and long implantation times. In a study by Bianco et al. about the effects of high surface area titanium implants in rabbits, the group was able to prove that for implants that do not undergo wear, there is still significant titanium release into the surrounding tissue: the mean local titanium content was almost twice as high for a 12-month implant than for the control experiments [Bianco et al., 1996]. This study proves that there is significant concern for the accumulation of titanium in tissue surrounding implants, even those that do not experience wear in physiological conditions.

Furthermore, the passivating oxide coating that forms on titanium is also susceptible to corrosion itself while protecting the metal surface underneath. The clinical concern with titanium and its alloys that do not undergo physical wear is associated with the breakdown of oxide layer and metal implant degradation [Manivasagam, 2010]. The native titanium oxide film contains a low density of microscopic sites that act as electrochemically active sites for soluble redox species including phosphate, chloride and bromide anions [Basame and White, 1999]. Because the films exhibit stable electrochemical redox activity and localized oxide breakdown, the system undergoes pitting corrosion to reveal the reactive titanium underneath [Basame and White, 1999]. Due to small variations in local oxide thickness and stoichiometry, fluctuations in electrochemical potential occur across the oxide surface, leading to pitting corrosion [Basame and White, 1999]. In physiological conditions, ion exchange occurs at the interface between the oxide and the physiological environment, leaving calcium and phosphate in the oxide surface [Chaturvedi, 2009]. As the oxide coating continues to transform, it degrades and reveals the underlying titanium, which can undergo galvanic corrosion in the fluids [Chaturvedi, 2009]. A safer implant with a longer useful lifetime can be created by further appending the oxide coating with a secondary passivation layer that could decrease the diffusion of ions between the titanium oxide coating and the surrounding biological conditions.

Studies of biomedical titanium are often performed in Ringer's Solution, a physiological saline solution containing sodium chloride, potassium chloride and calcium chloride.

Because Ringer's Solution contains chloride ions, they, rather than phosphate ions, provide the main pathway for the corrosion of titanium oxide. Pitting corrosion for metals in chloride solutions is unique because pits can nucleate and grow at potentials below the pitting potential [Burstein et al., 2004]. When chloride ions migrate and form metal chloride salts and the oxide-metal interface, the molar volume at that area increases and can cause additional rupture of the passivating film [Burstein et al., 2004]. These weakened points create nucleation sites for new pits in the oxide layer. Due to the combined effect of local concentrations in driving potential and the metal-chloride formation, Ringer's Solution is able to both simulate physiological conditions and provide an environment for titanium and titanium oxide corrosion.

Titanium oxide must also be protected from biofilm formation and bacterial adhesion for biomedical application. For dental implants, bacterial colonization along the transgingival region can be detrimental to the lifetime of the implant as the gums recede and increase bone area exposure to pathogens [Del Curto et al., 2005]. This bacterial growth can decrease osseointegration nearby and can affect a the growth of new bone [Del Curto et al., 2005]. In addition to the chemical properties of the adhesion surface, the topographical roughness also has a significant effect upon bacterial colonization. Surfaces with higher degrees of roughness show increased plaque colonization due to an increase in available area for the bacteria to grow and to the sheltering effect of irregularities that protect the bacteria from chemical and abrasive removal [Rimondini et al., 1997].

Dendrimers for Protection from Corrosion

Dendrimers are a class of structurally perfect branched macromolecules [Dykes, 2001]. As a result of the controlled synthesis in dendrimer formation, the final molecule is a highly branched, fractal polymer consisting of a core 'mer' and branches, or arms, extending outwards [Dykes, 2001]. The branches are formed by a series of polymerization reactions that add a monomer to the end of each arm, forming shells or generations. The first-generation refers to the first layer of branched units covalently bonded to the dendrimer core while each subsequent layer can be bonded after deprotection of the previous layer until the desired degree of branching is reached [Dykes, 2001]. There are three regimes where chemical interactions can occur within the dendrimer molecule: the encapsulated core, the cavities between the branches, and the multivalent surface [Dykes, 2001]. Depending on the the composition of the monomer and the generation of the dendrimer, the polymer will take a different shape in relation to the interacting electrostatic and steric forces. With strongly charged monomers and a high-generation polymer, the dendrimer molecules tend to adopt a spherical or globular shape which often results in the formation of internal cavities [Matthews et al., 1998]. These internal cavities have attracted attention as a possible means of trapping and transporting charged ions and molecules for drug

delivery purposes [Kaczorowska, 2009]. By extending this concept, applying a dendrimer monolayer to metallic biomedical implants could reduce the corrosion of the metal surface by decreasing ionic mobility between the metal surface and the physiological surroundings.

Because of the physical barrier, dendrimer coatings would act as diffusion barriers through steric hindrance of the tightly packed branches [Hiraiwa, 2006]. The dendrimers would not permanently trap the ions, as no irreversible structure would be formed between the metal ions and the surrounding dendrimer [Cheng et al., 2008]. Instead, the ions would have the physical barrier of the densely packed dendrimer branches to overcome that would reduce the rate of diffusion and corrosion. While some internal cavities within the molecule would be easier for the ions to pass through than others, the entire dendrimer would act as a periodic diffusion barrier where the diffusion coefficient for the ions at each position would depend on the dendrimer branch density at that point. As the polymer density varies radially as a function of distance from each dendrimer's center, the diffusion coefficient varies as well creating a diffusivity gradient along the molecule. While the coefficient may change, the molecule would still provide greater resistance for ion movement than a bare metal or oxide surface.

Due to the near-neutral biological solution the dendrimer would find itself in, the surface groups would be protonated, giving 64 positively-charged branch ends. This positive charge lends itself to metal ion attraction and containment. The counterion distribution and zeta-potential of the dendrimer extend beyond its radius of gyration, with G5 showing twice the effective distance as radius: 55Å as compared to a molecular radius of 25Å [Maiti et al., 2008].

A successfully formulated dendrimer film would have the potential to reduce the corrosion of metallic implants and reduce undesirable microbial activity to extend the lifetime of biomedical implants. One dendrimer type that would be capable of this is poly(amidoamine) (PAMAM) dendrimers, which are currently being researched for drug and gene delivery systems [Esfand et al., 2006]. As seen in Fig. 1, PAMAM dendrimers consist of an ethylenediamine core that undergoes repetitive alkylation and amination steps to produce each subsequent generation of the dendrimer [Esfand et al., 2005]. Previous research has been conducted into the chemical properties of PAMAM dendrimer films by Wang et al., examining the antimicrobial properties of dendrimer films on titanium oxide for biomedical implants [Wang et al., 2013]. After cleaning and oxidation of the titanium substrates, the samples were submerged in an aqueous dendrimer solution and incubated to form the coating. The coatings are not only biocompatible, but also antimicrobial, preventing the colonization and growth of two different bacteria strains even after 30 days of incubation. Because the PAMAM coating would prevent unwanted

growth of bacteria, it would benefit an implant by being both biocompatible and antimicrobial.

## Design Goals

The project's goal is to demonstrate that dendrimer films are capable of reducing corrosion of metallic implants, specifically titanium. The design, a monolayer dendrimer film on a titanium dioxide coated titanium substrate, was modeled using Kinetic Monte Carlo in MATLAB for film formation and diffusion physics. Additionally, a prototype of the film on a titanium oxide surface was created and characterized using atomic force microscopy and scanning electron microscopy. A successful design for a dendrimer film should inhibit the diffusion of titanium ions out of the surface or salt ions like calcium and chlorine into the oxide layer for corrosion.

Every titanium implant has a native oxide coating that occurs naturally in air. Many studies have attempted to control the oxide layer in order to passivate the titanium surface. There have also been studies on other surface modifications to enhance the passivation properties of titanium [Manivasigam, 2010]. The passivation of pure titanium can be specifically compared to studies of titanium surface corrosion and titanium ion release into biological solutions. Based on these studies, 0.350 mg/L of titanium ions can be released from pure titanium after 1 week of corrosion testing in salt solutions [Cortada et al., 2000; Josephs et al., 2009]. Alloys such as Ti-$Al_6$-$V_4$ show much lower levels of titanium ion release; however, aluminum and vanadium ions are still measurably released after 8 weeks of testing [Josephs et al., 2009]. Measuring an ion release of less than 0.350 mg/L over our testing period of 5 days was considered a successful design.

## Technical Approach

Kinetic Monte Carlo simulations have been selected as the best way to model the diffusion of dendrimers through the solution and their subsequent adhesion to the surface. KMC simulates diffusion of ions in a system using predetermined rates, or hopping probabilities. As our models assume no chemical interactions, relying instead on the energetics of diffusivities, dynamics were neglected. MATLAB was chosen as the software for our modeling efforts, based on ease-of-use and team familiarity with the program. Our team has segmented the computer simulation effort into two halves: modeling the diffusion of chlorine ions within the dendrimer (with the dendrimer adhered to the titanium oxide surface), and modeling the system on a broader scale, including multiple dendrimers with a predetermined surface coverage on the titanium oxide surface and constant ion concentration in the solution. The second simulation also provided the control, with the

dendrimer film removed. At present, research [Mansfield et al., 1995] has indicated that simulations of the actual structure of a dendrimer molecule are extremely complex and computationally expensive. Given our temporal restrictions and limited computational power, we have decided to model the dendrimers in two ways. For the simulation of a single dendrimer, the branches were predetermined and static. For the simulation of a dendrimer film, it was assumed the dendrimers were hard spheres.

## The Control Model

### *Simulation of Rutile Titanium Oxide Surface*

Simulation of the rutile titanium oxide surface was done under two assumptions. The first is that the solution containing the chlorine ions was infinite, with a constant concentration of chlorine ions. The second is that the titanium oxide was effectively infinite given the resolution of the simulation, and as such concentration gradients could be ignored. The simulation assumed a layer of titanium oxide, a second layer acting as the oxide-solution barrier, and the remaining layers acting as the solution.

### *Distribution of Dendrimers on Surface*

Under the assumption that the dendrimers will adhere to the surface as a serious of inflexible, non-diffusing spheres, the "hard ball" model can be adopted. Liquid crystal dendrimers have been shown to favor anisotropic A15 backing over both FCC and BCC lattices [Li et al., 2004]. A15 packing gives a surface coverage of 90.69%. Additional research [Bliznyuk et al., 1998; Mansfield et al., 1993] indicates the dendrimers "spread out" when adhered to a surface, a product of attraction to the substrate and repulsion between branches. The surface coverage given by A15 packing is thus a conservative estimate.

The dendrimer branches were predetermined and static throughout the simulation. The dendrimers have a relaxation time on the order of .1 microseconds for the fifth generation. As the films are manufactured prior to application, it can be assumed that no rearrangement will occur once in use. As the majority of branches of the dendrimer preferentially attach to the substrate, the model assumes that no surface rearrangement occurs [Mansfield et al., 1993].

The Dendrimer Film Model

*Part 1: Defining Global Variables and Creating the Initial System*

We begin the KMC simulation by defining certain global variables (of a constant value through each section of the code): these are the size of the total constraining volume (x, y, z), the number of chloride ions in the system (Particles), the number of time steps (stepmax), and the size of the dendrimer "unit cell" (a, b, c).

The initial system (VolMat) containing the dendrimer film and chloride ion solution is generated in two steps. First, an a x b x c matrix of zeros defining the size of the molecule is generated. The structure of the "unit cell" is then created by generating several a x b planes containing 2s in the location of the dendrimer branches and 0s in empty positions; these planes replace the planes of the zeros matrix and stack to emulate the structure of a single molecule (Figure 4). In this model, the dendrimers are approximated as roughly cubic, with truncated corners. The dendrimer film (DenLatt) is then approximated by tiling the unit cell across the xy size of the total constraining volume. Scripts are included that generate plots of both the individual molecule and the lattice of molecules. The film is inserted into the initial system by replacing c rows in the bottom of the total constraining volume with the layers of the tiled film.

The chloride ion solution is generated by creating an xy matrix of 1s with user-defined height z. To define the concentration of ions, which are represented in this simulation as 1s, we first characterize the numeric and subscript indices of each matrix entry. A random sampling of the defined number of 1s is then selected from the matrix, whose numeric and subscript indices are also determined. The indices of the selected 1s are compared to those of the entire matrix, and all indices not present in both sets are replaced by 0s, which represent open sites in this simulation. The solution matrix is then inserted into the initial system by replacing the rows above the dendrimer film in the total constraining volume.

Additional boundary conditions are applied before the diffusion simulation to ensure that no particles can be present at the edges of the system, which could cause issues with MATLAB attempting to scan a coordinate larger than the system dimensions. A script is included that generates a 3D plot of the initial system.

*Part 2: Defining Hopping Directions and Choosing a Particle to Move*

Particle selection begins by first assigning a new constraining volume, VolMat2, which is a 4D array that stores all the information of VolMat at each time step (step). After this definition, the largest and most important for loop of the code begins, which defines the

time step. At the end of every iteration of this loop, the resultant array (containing the new position of an ion that has diffused) is cycled back to the beginning of the loop, and a new ion is selected to hop.

An ion is randomly selected by finding all numeric indices of 1s in VolMat2 (in a given step) and placing them in a column vector. The iterative sum of all elements in that vector is taken and stored in a second vector of the same size; for example, if the initial vector is OnesVm and the sum of that vector is OneSum:

$$\textbf{(1)}\quad OnesVm = \begin{bmatrix} h \\ j \\ k \end{bmatrix} \quad OnesSum = \begin{bmatrix} h \\ h+j \\ h+j+k \end{bmatrix}$$

A random number r, where 0 < r < 1, is generated and multiplied by the last entry of OneSum, and the resultant number is compared to OnesVm. The first entry in OnesVm whose value is greater than or equal to the product of the random number and sum of indices is the index of the chosen ion, whose subscripts [I J K] are found and stored in a structure file (Position) as (xi, yi, zi). Possible directions for hopping in the total system lattice can now be defined relative to the selected ion's coordinates.

A while loop doubly reinforces the boundary constraints imposed previously and ensures that a particle cannot be selected for motion if it is at the edge of the system. The simulation will only proceed if the conditions specified in the while loop are met, ensuring that all selected ions cannot try to move to an index exceeding that of VolMat2.

There are two basic hopping directions in an empty solution: orthogonal and diagonal. The value (0, 1, or 2) of VolMat2 (the matrix describing the total system) must be determined in every possible hopping direction relative to the ion. A structure file called Direc is used to accomplish this task, which checks each of the 36 directions representing possible moves for the ion and stores them in a column vector. The basic orthogonal directions (up, down, left, right, out, in) and diagonal directions (upleft, upright, upout, upin, leftout, leftin, rightout, rightin, downleft, downright, downout, downin) are defined by adding or subtracting 1 from the appropriate coordinate(s) (xi, yi, zi) of the ion's position. Double hops in each direction are also considered (up2, down2, etc) and are characterized by adding or subtracting 2 for the appropriate coordinate(s). Each unique direction is associated with a specific hopping probability (Figure 5).

*Part 3: Diffusion of Chloride Ions and Calculation of Hopping Rates*

When considering the mass transfer of chloride ions from the solution to the surface of our device, the ions must travel through the dendrimer layer to reach the titanium surface. The

dendrimer consists of branches and voids, with different diffusivities or hopping rates associated with each. The total flux of ions through the dendrimer is therefore divided into two fluxes, weighted by their percent area, as seen in Eq. 2.

$$J = \frac{A_{br}}{A} J_{br} + \frac{(A - A_{br})}{A} J_{fl} \tag{2}$$

According to Fick's First Law, flux can be expressed as a diffusivity times a concentration gradient. Eq. 2 can therefore be replaced with Eq. 3. The concentration can be divided out, leaving an effective diffusivity relation, Eq. 4.

$$-\widetilde{D}\frac{dC}{dz} = -\frac{A_{br}}{A} D_{br} \frac{dC}{dz} - \frac{(A - A_{br})}{A} D_{fl} \frac{dC}{dz} \tag{3}$$

This effective diffusivity can be divided into diffusivities across branches or through voids of the dendrimer, weighted by the area taken by each dendrimer.

$$\widetilde{D} = \frac{A_{br}}{A} D_{br} + \frac{(A - A_{br})}{A} D_{fl} \tag{4}$$

Using the molecular dynamics data plotting counterion density vs. radial distance from Maiti et al., the number of counterions was counted at each distance. [Maiti et al., 2008] This plot is provided in Fig. 2. Dividing each number by the total number of counterions provided a probability of finding a counterion at a given distance from the center of the dendrimer. Equating this probability to a term for energy gave an approximate energy associated with that position, as seen in Eq. 5 and Eq.6.

$$P(r) = v * e^{(-E/kT)} \tag{5}$$

$$E = -kT * \ln\left(\frac{P(r)}{v}\right) \tag{6}$$

The time step from the molecular simulations, 2 fs, was selected as v. These energies allowed the determination of a change in energy associated with moving through the dendrimer. The dendrimer was divided into layers, based on generation and monomer density. This monomer density was determined from the monomer density vs. radial distance from molecular dynamics. For each layer, the effective diffusivity was determined based on the relative energy previously determined using Eq. 7.

$$\widetilde{D} = D_0 e^{(-E/kT)} \tag{7}$$

To determine $D_0$, it was assumed that at infinite temperatures ions would move past branches as if they were moving through fluid. Consequently, the diffusivity of ions in water was taken to be $D_0$.

The $D_{fl}$ corresponds to the $D_0$ used previously. $D_{fl}$ was determined from a study by Li and Gregory on the diffusion of chloride ions through salt solution [Li and Gregory, 1974]. The

diffusivity was plotted as a linear function of temperature, as seen in Fig. 3, and was extrapolated to 37 C, the temperature of our corrosion tests, to determine $D_{fl}$.

The total area (A) of the layer was previously determined, when the dendrimer was divided up into layers based on generation and monomer density. The area covered by branches in each layer was determined by calculating the number of monomers/branch, the number of monomers/layer, and the approximate area of a monomer based on covalent radii of the appropriate elements.

This approximate monomer area is based on that of ethylenediamine. Its widest part is the $CH_2$ section, approximately 1.35Å wide. Its longest part is the N-C-C-N backbone, approximately 2.88Å long. This gives a final approximate rectangular area of 3.89 $Å^2$.The $D_{br}$ can be determined by calculation.

$$D_{br} = \frac{\widetilde{D} - \frac{(A - A_{br})}{A} D_{fl}}{\frac{A_{br}}{A}} \tag{8}$$

$$D_{br} = \left(\frac{1}{6}\alpha^2 \upsilon e^{\left(-E_A^{br}/kT\right)}\right) \tag{9}$$

The lattice parameter (alpha) was taken as 1 Å. This final equation could be adjusted to determine hopping rate, instead of diffusivity.

$$\Gamma_{br} = \upsilon e^{\left(-E_A^{br}/kT\right)} = \frac{6D_{br}}{\alpha^2} \tag{10}$$

From this treatment, we have determined that the chloride ions diffuse much more slowly into and out of the dendrimer than through solution. However, once inside the dendrimer the ions diffuse much more quickly. This treatment yields hopping rates of $7.82 \times 10^{-5}$ 1/fs for ions diffusing through the solution, $3.69 \times 10^{-5}$ 1/fs for ions diffusing from the solution over the surface branches of the dendrimer, and $1.68 \times 10^{-3}$ 1/fs for ions diffusing across internal branches of the dendrimer.

The calculations indicate that diffusion within the dendrimer is faster than diffusion in the solution. This result is counterintuitive, as the dendrimer consists of solution diffusion as well as steric barriers. We attribute this difference to the source of the counterion density, which considered distribution at equilibrium. At near-equilibrium conditions, ion diffusion between voids favors hopping across dendrimer branches and produces a higher diffusivity and a higher effective diffusivity

*Part 4: Defining Hopping Probability in Different Directions*

Hopping probability in a given direction is related to two important parameters: the occupancy of a site and the site's relative position to the selection ion. From the calculation of hopping rates for chloride ion diffusion, a hopping probability for a single hop in each orthogonal direction ($\pm$x, $\pm$y, $\pm$z) was determined, as well as the range of hopping probabilities for ions within the dendrimer molecules. To define the hopping probabilities in the remaining 30 directions (double orthogonal, single diagonal, and double diagonal hops), a geometric approach was used to estimate the probability of a hop in a direction relative to the probability of a single orthogonal hop.

In accordance with Kinetic Monte Carlo theory, the constraining volume is emulated as a lattice occupied by 2s (representing components of dendrimer branches), 1s (chloride ions), and 0s (open "sites" in the solution). Movement of an ion to a new location can be viewed as a particle in a lattice migrating to a new position by stretching, compressing, and breaking the "bonds" it had with its preliminary neighbors (Figure 6). This allows treatment of the bonds as simple harmonic oscillators, where the potential energy barrier associated with each hop can be approximated with Hooke's Law using the spring constant and the cumulative sum of the changes in the lengths of all bonds associated with an ionic movement. Assuming that the transition state of the ion between its initial and final positions represents the most energetic transition state (and thus the potential barrier to motion in that direction), the potential barriers for motion in each of the 4 basic hopping cases can be written:

*Orthogonal single hop*

$$\Delta E_D = \sum_i^N \frac{k}{2}(\Delta L_i)^2 = \frac{k}{2}[(2-1)^2 + 4(\sqrt{2}-1)^2]$$

$$\Delta E_D = k[\frac{13}{2} - 4\sqrt{2}]$$

$$\Delta E_D \cong 0.843$$

(11)

*Orthogonal double hop*

$$\Delta E_D = \sum_i^N \frac{k}{2}(\Delta L_i)^2 = \frac{k}{2}[2(\frac{\sqrt{5}}{\sqrt{2}} - 1)^2 + 2(1 - \frac{1}{\sqrt{2}})^2 + 2(\frac{\sqrt{3}}{\sqrt{2}} - 1)^2]$$

$$\Delta E_D = k[\frac{15}{2} - \sqrt{2}(1 + \sqrt{5} + \sqrt{3})]$$

$$\Delta E_D \cong 0.474$$

*Diagonal single hop*

$$\Delta E_D = \sum_i^N \frac{k}{2}(\Delta L_i)^2 = \frac{k}{2}[2(\frac{\sqrt{5}}{\sqrt{2}} - 1)^2 + 2(1 - \frac{1}{\sqrt{2}})^2 + 2(\frac{\sqrt{3}}{\sqrt{2}} - 1)^2] \tag{13}$$

$$\Delta E_D = k[\frac{15}{2} - \sqrt{2}(1 + \sqrt{5} + \sqrt{3})] \tag{14}$$

$$\Delta E_D \cong 0.474$$

*Diagonal double hop*

$$\Delta E_D = \sum_i^N \frac{k}{2}(\Delta L_i)^2 = \frac{k}{2}[2(1-1)^2 + 2(\sqrt{5}-1)^2 + 2(\sqrt{3}-1)^2] \tag{15}$$

$$\Delta E_D = k[10 - 2\sqrt{5} - 2\sqrt{3}] \tag{16}$$

$$\Delta E_D \cong 2.064$$

Using a single orthogonal hop as a standard reference, the relative likelihood of a double orthogonal, single diagonal, and double diagonal hop can be approximated. This is the technique used to assign probabilities in the MATLAB model, where single- and double-orthogonal hops are denoted *pFluid* and *pFluid2*, and single- and double- diagonal hops are denoted *pDiag* and *pDiag2*.

*Part 5: Using Probability to Choose a Direction and Moving the Selected Ion*

For the entire system, there are two classes of ion hopping (diffusion): in the solution and in the dendrimer film or at its boundary. Orthogonal and diagonal hops are always possible, while branch hops (across the branches of the dendrimer molecule) are only possible for ions attempting to enter, diffuse within, or leave the dendrimer film (Figure 7).

Orthogonal hopping is more likely than diagonal hopping, while the likelihood of branch hopping increases with proximity to the center of a dendrimer due to the inverse relation between branch density and distance from center.

For all directions, there are five basic cases to be considered before the hopping probability in a direction can be assigned. Each basic case for a single and double hop in a given direction is illustrated in Figure 8.

Once the probability of a hop in each of the 36 total possible directions has been assigned, it is possible to select a hopping direction and move the ion. The probabilities are stored in a 36 by 1 column vector k , whose iterative sum is taken and stored in another column vector of the same size. A random number r between 0 and 1 is multiplied by the last entry of k, and the resultant value is compared to the vector of iterative sums, and the index of the first value greater than or equal to is chosen. The index of that value is between 1 and 36, and directly corresponds to the direction in which the ion will be moved. This method of randomly selecting a direction from a range of finite probabilities has several advantages. Primarily, directions that are closed- namely, those containing a 1 (ion) or 2 (portion of a dendrimer) – cannot be selected because they must always correspond to an index whose value is nonzero. Additionally, the likelihood of a direction being chosen scales by the probability of a hop in that direction. A very small value of p corresponds to a narrow range of the cumulative sum of k, and the direction is less likely to be chosen in comparison to a larger value of p (and therefore range in k).

A second structure file containing positions relative to the selected ion is created (Direc2). This file is identical to the first, but also incorporates a time step. For every time step, a certain index (between 1 and 36) of the cumulative probability vector is chosen, which corresponds to a particular direction of movement (1 = up, 2 = down, etc). A series of if and elseif statements change the value of initial location of the ion (xi, yi, zi) in VolMat2 from 1 to 0, and the value of the direction in which it moves from 0 to 1. The new matrix containing the modified position of the selected ion is cycled back to the beginning of the for loop defining the time steps, and the process repeats until all time steps have been completed.

While the ions are diffusing, a plotting script finds and counts all 1s (corresponding to ions) in the z-ranges of the matrix corresponding to the dendrimer film and the open space below it, and plots the values per each time step. The plot displays the net ion count inside the film and the net ion count that diffused through the film as a function of time (Figure 9).

Advantages and Disadvantages

The KMC model described above has the advantage of being simple in theory. This enables emulation of an effective dendrimer film without having to recreate a complex fractal construct, which would be both computationally difficult and expensive in MATLAB. However, the simplicity of the model is also not realistic; obviously, the system is not a lattice, and fluidic diffusion is not limited to a set number of directions and corresponding "open sites". The dendrimers are not all identical or positioned in a tiled array, and this structure is many times more complex than a roughly truncated cube.

Another drawback of this technique is the duration of time required for completion of a meaningful number of time steps. Each step takes much longer in real time than the femtoseconds it represents, and a simulation of the real-time duration of testing (5 days) would require much more time than we have allotted for this project. Additionally, the concentration of chloride ions in the actual solution is very low, and would translate to approximately 2 ions per layer of the constraining matrix. Given the time required for simulation, it would be difficult to obtain meaningful results using such a low concentration, so it is necessary to input a much higher chloride concentration than the real solution contains.

Nevertheless, even the results of these very basic approximations of the system's dynamics yield useful information about ion diffusion through the dendrimer film (see Discussion and Results.) 1000 iterations corresponds to 2 picoseconds.

System Model

For the control system, the dendrimer film was removed, leaving a system with a titanium oxide substrate and solution containing chlorine ions. Modeling was done with kinetic Monte Carlo, similarly to the dendrimer model. Using adsorption, surface diffusion, and bulk diffusion values derived from density functional theory simulations [Inderwildi et al., 2008], hopping probabilities were calculated using

$$P_i = \frac{\nu * e^{-\Delta G_i / RT}}{\sum_{1}^{n} \nu * e^{-\Delta G_i / RT}} \tag{17}$$

The bulk titanium oxide was given a z-coordinate of 1; the surface a z-coordinate of 2; the solution began at z-coordinate of 3. The concentration of chloride ions was kept constant in the solution and concentration gradients were not taken into consideration, as the model assumed infinite titanium oxide and solution. A chlorine ion was chosen at random from all ions either in solution or adhered to the surface. The system had no adhered ions at zero time. Hopping probabilities were considered in six directions, positive and negative along each of the three axes. The model considered three distinct possibilities, depending on the z-coordinate of the chosen ion. If the ion was at z-coordinate 4 or higher, jumps in any direction not occupied by another ion were given equal probability. For an ion at z-coordinate 3, the probability of a downward hop was dictated by the energetics associated with chlorine adhesion onto titanium oxide; all other directions were considered equal, with hopping probability dictated by diffusion of chlorine in solution. For an ion at z-

coordinate 2, three options were possible: diffusion along the surface, diffusion into the bulk, and desorption from the titanium oxide surface; hopping probabilities were calculated given the energetics of each.

## Prototype

To determine experimentally the ability of a dendrimer monolayer to retard ion diffusion between a physiological solution and a titanium substrate, a prototype titanium sample with an oxide surface and a dendrimer film was fabricated and tested. The titanium substrate was deposited on a silicon wafer by sputtering and oxidized by either plasma oxidation or thermal oxidation. The dendrimer film was then fabricated in methanol solution and tested in Ringer's solution to determine the diffusion of titanium out of the substrate and the diffusion of chlorine and calcium into the oxide. The prototype was imaged using SEM and atomic force microscopy (AFM), then characterized by ellipsometry and inductively-coupled plasma optical emission spectrometry (ICP-OES) to determine the differences in diffusivity between coated and uncoated samples.

### Materials

The materials and laboratory space needed for prototyping were purchased through the Materials Science and Engineering department at the University of Maryland. A silicon wafer of 4-inch diameter and pure titanium sputtering targets were accessed through the UMD Nanocenter FabLab. The concentrated solution of G5 PAMAM dendrimer with amine functionalization in methanol was purchased from Sigma-Aldrich. The salts necessary for Ringer's solution test were donated by Dr. Seog and Dr. Briber. The titanium standard for ICP-OES measurements was purchased from SPEX Certi-Prep.

### Methods

Device samples and materials were stored at the Polymer Science Laboratory. Silicon substrates of 10x10mm were rinsed with acetone, methanol, isopropanol and deionized water and dried with N2 gas to remove any potential contaminants on the substrates in the UMD Nanocenter Fabrication Laboratory (UMD FabLab). Titanium samples were prepared by sputtering pure titanium onto these silicon wafer pieces. Silicon was chosen as the substrate material because it provided an atomically flat surface for the sputtered titanium film to deposit onto. Additionally, according to Yu et al., a thin 8nm amorphous region forms at the titanium/silicon boundary during sputtering that promotes adhesion of the two materials [Yu et al., 2007]. The sputtering was performed using the AJA Sputtering unit at the UMD Nanocenter FabLab for 200 minutes at a deposition rate of 50 nm/min under 200W and 5 mtorr of pressure to create a 1 um thick titanium layer.

The sputtered samples were separated into three groups: plasma oxidized, thermally oxidized, and bare titanium samples. Some titanium-coated samples were oxidized in the sputtering chamber at the FabLab by plasma oxidation in pure $O_2$ environment at atmospheric pressure at 400 C for 2 minutes to create a oxidized titanium surface. This procedure was based off of a study by He et al., which used a similar method to generate an amorphous oxide layer on sputtered titanium in a sputtering chamber [He et al., 2004]. Other samples were oxidized using in the annealing furnace at the FabLab for 1 hour at 700°C in atmospheric conditions to create a rutile surface oxide. The oxide and titanium surfaces were imaged using SEM, AFM and ellipsometry to determine the characteristics of the oxides.

The oxidized titanium samples were then coated in a dendrimer monolayer based on the procedure by Wang, et al. [Wang, 2013]. Firstly, a 0.39 mM dendrimer solution in methanol was prepared by diluting the concentrated solution from Sigma-Aldrich with 2.71mL of methanol for every 1 mL of concentrated solution. The oxidized samples were immersed in this solution for 2 hours with a rate of shaking at 45rpm and then let air dry. Wet lab procedures and preparation of this solution were performed in the Polymer Science Laboratory with permission from Dr. Briber.

Ringer's solution was used to test the corrosion of the titanium samples in physiological conditions. The solution was made using 7.2 g/L NaCl, 0.37 g/L KCl, 0.17 $CaCl_2$ with a total pH of ~7 [Helmenstine, 2013]. The 100 mL of solution was prepared for use in experiments and pH was confirmed using pH strips. Each sample was placed in its own glass bottle containing 10 mL of Ringer's solution at a temperature of 37°C for 5 days.

## Ethical and Environmental Impact

The overall goal of the project was to create a final product that would benefit society by improving the long term reliability of biomedical implants. The materials we used, specifically the PAMAM dendrimer films and the titanium used as a substrate, are biocompatible. According to the PAMAM MSDS available from Dendritech, polyamidoamide is a non-hazardous material. [Polyamidoamine, 2011] However, since the PAMAM was suspended in methanol, which is considered toxic [Methanol, 2013] fume hoods, latex gloves, eye protection, and lab coats were used when handling the solution. Any waste containing methanol was disposed of in the appropriate waste containers supplied by the Polymer Science Lab. The waste from the corrosion testing was composed primarily of salts, with extremely small concentrations of PAMAM and titanium; as such, it could be disposed of in any organic solution in the Polymer Science Lab and posed little risk to the environment.

There is substantial potential gain from the use of dendrimer films as a protecting film on medical implants. By inhibiting corrosion both into and out of the implant, the possibility of degradation of the implant and contamination of surrounding tissue would be decreased, which in turn extends the life of the implant and increases the quality of life for the patient. One concern is the long-term stability and toxicity of dendrimers in the human body and the effect of the different functionality of the dendrimer on cell surface interaction [Duncan and Izzo, 2005]. Few toxicology experiments have been performed so far, and thorough in vivo and in vitro experiments must be performed before a dendrimer monolayer coating for implants would be usable.

## Intellectual Merit

Our project deals primarily with diffusion and corrosion mechanisms of titanium in physiological solutions and the influence of dendrimers on the diffusion process. Based on the technical background there is very little literature on the diffusivities of physiologically relevant ions through polymer films or other implant materials. The design and simulations of this project will provide sorely needed insights into the interactions between salt ions and passivation structures such as dendrimers. Our prototyping and testing results will also provide an initial proof-of-concept of a dendrimer corrosion passivation technique. Further testing will be required to determine if this solution is viable for long-term implant corrosion passivation, but this is an important first step for addressing this problem in the biomedical field.

## Broader Impact

By developing a biocompatible and corrosion resistant coating for titanium and titanium oxide surfaces, a new passivation layer can be created for metal biomedical implants. The useful lifetimes of current biomedical implants are limited by the degradation of the material within the body and the reduction of mechanical properties of the device. A dendrimer coating could be used to extend the lifetime and reliability of implants that do not undergo wear by decreasing the rate degradation of the exposed titanium surfaces.

## Results and Discussion

### Model Simulation Results

Several simulations were performed using this model, for durations of 2000 time steps. The simulated area was 24 x 24 nm, and the solution volume was 2765 $nm^3$. The solution was defined to contain 3,000 ions, for a net molarity of 1.80 x $10^{24}$ $M*nm^{-3}$. All simulations

showed ion diffusion into the dendrimer film, with ion counts in the film ranging from 10-25 (Figure 8). None of the simulations showed any ion diffusion *through* the film, however, indicating that the dendrimer film successfully impeded ion penetration into the space below the film (and, by extrapolation, a titanium substrate). These simulations give an average flux of $1.22*10^{28}$ $m^{-2}$ $s^{-1}$ of ions into the dendrimer film, but an effective flux of 0.00 $m^{-2}$ $s^{-1}$ of ions from the dendrimer film into the titanium oxide surface.

System Model Simulation Results

Given $10^6$ iterations of the model, which simulates $2*10^{-9}$ seconds, 1761 ions diffused giving a flux of $3.52*10^{26}$ $m^{-2}$ $s^{-1}$ as seen in Fig. 10. This is a conservative result, as the model did not account for diffusion out of the titanium oxide; the actual flux is lower.

Deposition of Titanium

Titanium deposited on a silicon substrate forms a Si/Ti layer to promote adhesion between the silicon wafer and sputtered titanium [Yu et al., 2007]. According to Yu et al. and their procedure for titanium deposition in a sputtering chamber, there is a 8nm amorphous layer at the Si/Ti interface as seen in Fig. 11. Additionally, the interface generated does not undergo significant changes after further annealing of the samples [Yu et al., 2007].

The roughness of our deposited titanium was determined by AFM. As seen in Fig. 12, the roughness of the sputtered titanium is 5.605 nm. These AFM images were taken to compare with our SEM data to provide confirmation of the features.

Oxidation of Titanium

SEM and AFM were used to image the surfaces of the titanium oxide coatings as seen in Fig. 13, Fig. 14, and Fig. 15. The difference in surface properties between the thermal oxidation and the plasma oxidation is a product of the style of oxidation. The thermally annealed oxide has rougher overall surface due to the presence of grains in the rutile structure. The RMS roughness calculated by the AFM was 10.497 nm, seen in Fig. 14. The plasma oxide EDS in Fig. 16 was used in conjunction with SEM to prove the distribution of oxygen and titanium on the surface of the samples. The plasma titanium oxide AFM images calculated an RMS roughness of 9.227 nm, seen in Fig. 15.

There are many articles dealing with the effects of different titanium allotropes on cell adhesion and integration for biomedical implants [Sebbowa et al., 2011; Turzo, 2012; Vandrovcová and Bačáková, 2011; He et al., 2008; Sul et al., 2002; Huang et al., 2003]. For our prototyping, we used rutile and amorphous titanium oxide surfaces for our oxide

coatings. The rutile conformation is the most energetically favorable of the allotropes. It is formed by high temperature annealing of the oxide or by oxidation at temperatures above 700°C [Ting and Chen, 2000]. Annealing at temperatures below 700°C produces either a purely anatase or a mixed oxide coating. Amorphous titanium oxide is formed during oxidation at lower temperatures below 550°C for short periods of time where the oxide does not have the time or thermal energy to diffuse into its lowest energy state [Ting and Chen, 2000].

Our oxidation techniques provided us with rutile and amorphous final products. As demonstrated by Wei-Feng et al. and Ting et. al, and as seen is Fig. 17, thermal oxidation of oxide at 700°C will generate a purely rutile titanium oxide layer [Ting and Chen, 2000; Wei-Feng et al., 2008]. Therefore, the samples that were oxidized in the annealing furnace at 700°C should have a purely rutile structure. SEM images of the thermally oxidized samples can be seen in Fig. 13a.

On the other hand, the samples oxidized by plasma oxidation in the sputtering chamber should have an amorphous structure based off the results of Droulers et al. [Droulers et al., 2011]. The plasma chamber was at a low enough temperature that the final structure will be amorphous when cooled. The SEM images in Fig. 13b show the morphology of the 45 degree edge of the plasma oxidized samples. The pits visible in the image originate from the nature of high energy plasma ions.

Characterization of Titanium Oxide Thickness

Ellipsometry was used to determine the thickness of the oxide layers for the two different oxidation procedures. The Woollam M-2000 ellipsometer in the IREAP labs was used for this characterization. Based on IREAP's fitting models, this measurement determined that the plasma-oxidized sample had an oxide layer approximately 1 nm thick. It determined that the thermally oxidized sample had an oxide layer approximately 1600 nm thick. This implies that nearly all of the titanium was oxidized by thermal oxidation while only a small fraction was oxidized by plasma oxidation. The ellipsometry data was plotted as Psi vs. wavelength, with Psi being a function of light polarization and incident angle. These plots can be seen in Fig. 18.

Characterization of Dendrimer Film

After the dendrimer films were generated, SEM in conjunction with EDS was used to determine the chemical composition of the films on the samples as seen in Table 1. The EDS results of the SEM images reveal small amounts of nitrogen and carbon on the titanium oxide surface. Before film deposition, the only elements present in the samples were

titanium, oxygen and silicon, depending on the depth of EDS penetration. However, after the films were applied, there were small amounts of nitrogen and carbon detected signifying the presence of the organic PAMAM dendrimers on the titanium oxide surface.

Measurement of Surface Corrosion and Ion Diffusion

Diffusion of ions out of the film and into the surrounding solution is is a critical parameter characterizing the efficiency of our device. To compare the ion diffusion of an oxide-only coated titanium substrate and an oxide and dendrimer coated sample, one of each type of sample was submerged in separate Ringer's solutions for 120 hours (5 days). The samples were then removed.

The resultant titanium concentrations in each solution were measured and compared. The concentration of metallic ions in solution was measured by ICP-OES at the UMD Laboratory for Advanced Materials Processing with permission from Dr. Rubloff and the help of Jonathon Duay. Because the salt compounds in the Ringer's solution can clog the spectrometer's nebulizer, the experimental solutions were diluted by a factor of 2. The values detected by spectroscopy were compared to a standard curve, based on a titanium standard purchased from Sigma-Aldrich, to determine the final concentration of titanium in the solution. The ion concentration measurement was repeated three times by the device and the average ion concentrations were calculated. Based on a linear fit of the standard curve, the bare titanium surface released enough titanium ions to produce a concentration of 0.58 ng/L. The plasma-oxidized surface released approximately 52% fewer ions, or 0.28 ng/L. The thermally oxidized surface released approximately 68% less, or 0.19 ng/L. The plasma oxidized and dendrimer functionalized surface released approximately 67% fewer ions, or 0.18 ng/L. The thermally oxidized and dendrimer functionalized surface released approximately 61% less, or 0.22 ng/L. This data is plotted in Fig. 19. These concentrations in relation to the standard curve, and the fitting parameters used, are plotted in Fig. 20. Based on the differences in ionic concentrations, the ICP-OES results prove that the oxide and the dendrimer coatings reduced the amount of corrosion occurring in the device. The dendrimer film decreases the amount of corrosion occurring when adhered to a plasma-oxidized device, however it appears that the film does not reduce a thermally oxidized device's corrosion.

Additionally, the samples were imaged by SEM and analyzed using EDS after incubation in solution. Fig. 21a and b show the thermally oxidized samples with and without dendrimer coatings after incubation in solution. As seen in Fig. 21 and Fig. 22, the thermally oxidized samples fractured after incubation that may be due to reduced adhesion between the silicon wafer and the titanium oxide surface after oxidation. In Fig. 22 and Table 2, the EDS results reveal the surface under the crack is the silicon wafer. While the silicon/titanium

interface is well established, the silicon/titanium oxide is weaker and may be susceptible to cracking. Therefore, titanium oxide surfaces should be directly formed on the deposited titanium to prevent delamination.

Fig. 23 shows the samples prepared by plasma oxidation were imaged after incubation and were not significantly different than the samples before incubation. Because of the insulating ability of the organic PAMAM dendrimers, the SEM images for the dendrimer samples are less clear than their bare counterparts.

The respective EDS for the thermally oxidized samples and the plasma-oxidized samples can be seen in Table 3. The elemental analyses of the samples reveal that the dendrimers are still present on the surface, based upon higher nitrogen and carbon content in the samples coated by the film compared to the bare samples. Additionally, the dendrimer-coated samples had a higher concentration of chlorine on the surface than the bare samples. This difference in concentration is due to the ability of the dendrimer film's propensity for ion trapping: when the samples were washed after incubation, chloride ions would remain trapped in the dendrimer branches while all of the chlorine on the bare samples would be washed away in solution.

Based on the results of the prototype testing, electrostatically deposited dendrimer layers on titanium oxide can decrease the corrosion rate of the underlying bulk titanium. The ICP-OES data shows there is significantly less titanium release by oxidized and dendrimer coated samples than bare samples when exposed to physiological solution. The ICP-OES results showed the thermally oxidized samples released the lowest number of ions into solution; however, SEM and EDS characterization revealed that these films cracked and therefore had poor adhesion capabilities. The plasma-oxidized surface coated with dendrimer released the fewest titanium ions while maintaining film adhesion.

The design and modeling of the titanium-dendrimer system implied that a dendrimer layer would decrease the amount of chloride ions incident on the titanium oxide surface. This diffusion barrier would decrease the amount of corrosion due to the chloride concentration gradient that causes pitting corrosion. Our prototype testing results uphold this initial model, showing that dendrimers do in fact reduce the corrosion of the titanium surface.

## Conclusions and Future Work

Due to titanium and titanium oxide's corrosive susceptibility in physiological environments, additional passivation of biomedical implants serves the highly useful purpose of extending their useful lifetimes and safety. Based on modeling and experimental results, PAMAM dendrimers can successfully reduce the corrosion rate of an oxide-coated titanium surface.

The densely packed branches of the dendrimers act to impede diffusion of the corrosion-inducing chloride ions from the surrounding solution to the oxide surface. According to the KMC MATLAB simulation, the dendrimers trap chloride ions within their branches based on low hopping probabilities in and out of the densely packed outer layer. Prototyping of the dendrimer film was accomplished using sputtered titanium samples that were oxidized and coated with a PAMAM dendrimer. The EDS analysis demonstrates that the dendrimers successfully trap the chloride ions, while the ICP-OES results show increased diffusion in titanium surfaces without dendrimer coating. In summary, a PAMAM dendrimer coating can lower the degree of pitting corrosion of titanium implants by decreasing the concentration of reactive chloride ions at the interface between the titanium oxide coating the physiological solution.

Future Design Work

The Kinetic Monte Carlo simulations require a much longer period of time to run to completion than the time available for this project. They could also be made more statistically useful with a larger amount of memory available. Design parameters could be made clearer by optimizing these simulations.

Enhanced complexity in the system would also increase the accuracy of the simulations. Due to the time constraints imposed upon this project, many assumptions were made in the construction of the simulation that deviated from reality, like the shape and distribution of the dendrimers in the dendrimer film. A more spherical dendrimer with more precisely defined branches would better help to simulate ion diffusion through the film, as would an array that more closely emulated the surface packing of the molecules on the substrate.

Future Prototyping Work

X-ray photoelectron spectroscopy (XPS) could be used to analyze the chemical composition of the surface of the prototyped samples. By measuring the energy of the photoelectrons released by samples when irradiated with soft x-rays, elemental analysis of the surface of samples can be performed. XPS could help analyze the chemical composition of the titanium oxide after corrosion testing of our samples. The elemental analysis would reveal chlorine and calcium that has diffused through the top layers of the titanium oxide samples that would cause degradation of the protective oxide. By comparing the elemental composition of dendrimer coated and bare samples after corrosion testing, more substantial results can be collected about ion diffusion into the samples.

Electrochemical impedance spectroscopy (EIS) is a commonly used technique for studying corrosion of films or at surfaces [Randviir, 2013]. For our project this technique could give us a corrosion rate and change in layer thickness of both the dendrimer layer and the oxide layer. Producing these numbers would require developing a model of the electrical properties of these films. Modeling titanium oxide would be a simple step, however, modeling the electrical properties of the dendrimer film could pose a significant problem. Models of other polymer systems could provide guidance about how best to approach this issue, but significant background research would be required in order to determine appropriate assumptions to make.

A study of the adhesion properties of the dendrimer film is relevant to its use as a biomedical implant film. Most implants undergo fatigue over their entire lifetimes, so delamination could be a significant problem for this dendrimer film. The current design relies on electrostatic attraction between the film and titanium oxide surface, which is known to be a relatively weak interaction. There has been work on covalent bonding of dendrimers to titanium [Wang, 2011]. Covalently bound films would be much more mechanically stable; however, films bound in this fashion necessitate a more complicated fabrication process and would also require a more complicated model to study diffusion through the barrier.

**Acknowledgements**

Special thanks to:
Dr. Phaneuf, Dr. Briber, Dr. Zhang, Dr. Seog, Nitinun Varongchayakul, Dr. Rubloff, Jonathon Duay, Alex Kozen, Tom Loughran, Jon Hummel, UMD Nanocenter Fabrication Laboratory

**Tables:**

Table 1: The EDS analysis of the thermally oxidized titanium samples (a) and the plasma oxidized samples (b) after the dendrimer film had been applied.

a)

Spectrum: _1 518

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Ti | 22 | K-series | 58.42 | 59.51 | 32.44 | 1.8 |
| N | 7 | K-series | 3.86 | 3.93 | 7.32 | 6.5 |
| C | 6 | K-series | 1.04 | 1.06 | 2.31 | 0.3 |
| O | 8 | K-series | 34.85 | 35.50 | 57.92 | 40.9 |
| | | Total: | 98.18 | 100.00 | 100.00 | |

b)

Spectrum: _1 541

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Ti | 22 | K-series | 72.48 | 72.73 | 45.55 | 2.2 |
| N | 7 | K-series | 9.27 | 9.30 | 19.90 | 13.3 |
| C | 6 | K-series | 2.53 | 2.54 | 6.35 | 0.5 |
| Si | 14 | K-series | 0.88 | 0.88 | 0.94 | 0.1 |
| O | 8 | K-series | 14.50 | 14.55 | 27.26 | 24.0 |
| | | Total: | 99.66 | 100.00 | 100.00 | |

Table 2: EDS results for the surface (a) and the open crack (b) for the thermal oxide samples without dendrimer coating after incubation.

a)

Spectrum: _1 528

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Ti | 22 | K-series | 55.68 | 58.55 | 31.75 | 1.7 |
| N | 7 | K-series | 4.07 | 4.28 | 7.94 | 6.9 |
| C | 6 | K-series | 0.51 | 0.53 | 1.16 | 0.2 |
| Cl | 17 | K-series | 0.21 | 0.22 | 0.16 | 0.0 |
| Si | 14 | K-series | 0.14 | 0.14 | 0.13 | 0.0 |
| O | 8 | K-series | 34.48 | 36.27 | 58.86 | 44.2 |
| | | Total: | 95.09 | 100.00 | 100.00 | |

b)

Spectrum: _1 527

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Si | 14 | K-series | 88.83 | 90.21 | 82.90 | 3.7 |
| N | 7 | K-series | 4.64 | 4.72 | 8.69 | 8.3 |
| C | 6 | K-series | 1.97 | 2.00 | 4.29 | 0.9 |
| Ti | 22 | K-series | 0.56 | 0.57 | 0.31 | 0.1 |
| Cl | 17 | K-series | 0.26 | 0.26 | 0.19 | 0.1 |
| O | 8 | K-series | 2.21 | 2.24 | 3.62 | 3.8 |
| | | Total: | 98.47 | 100.00 | 100.00 | |

Table 3: The EDS results after the incubation experiment for the bare (a) and dendrimer coated (b) thermal oxidized samples and for bare (c) and dendrimer coated (d) plasma oxidized samples.

Spectrum: _1 528

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Ti | 22 | K-series | 55.68 | 58.55 | 31.75 | 1.7 |
| N | 7 | K-series | 4.07 | 4.28 | 7.94 | 6.9 |
| C | 6 | K-series | 0.51 | 0.53 | 1.16 | 0.2 |
| Cl | 17 | K-series | 0.21 | 0.22 | 0.16 | 0.0 |
| Si | 14 | K-series | 0.14 | 0.14 | 0.13 | 0.0 |
| O | 8 | K-series | 34.48 | 36.27 | 58.86 | 44.2 |
| | | Total: | 95.09 | 100.00 | 100.00 | |

a)

Spectrum: _1 534

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Ti | 22 | K-series | 54.31 | 57.86 | 31.09 | 1.7 |
| N | 7 | K-series | 3.89 | 4.14 | 7.60 | 6.6 |
| C | 6 | K-series | 0.72 | 0.77 | 1.65 | 0.3 |
| Si | 14 | K-series | 0.16 | 0.17 | 0.16 | 0.0 |
| Cl | 17 | K-series | 0.10 | 0.11 | 0.18 | 0.0 |
| O | 8 | K-series | 34.68 | 36.95 | 59.31 | 44.5 |
| | | Total: | 93.86 | 100.00 | 100.00 | |

b)

c)

Spectrum: _1 537

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Ti | 22 | K-series | 68.51 | 72.05 | 45.26 | 2.1 |
| N | 7 | K-series | 9.95 | 10.46 | 22.47 | 13.8 |
| Si | 14 | K-series | 1.10 | 1.16 | 1.24 | 0.1 |
| C | 6 | K-series | 0.56 | 0.59 | 1.48 | 0.2 |
| Cl | 17 | K-series | 0.04 | 0.04 | 0.03 | 0.0 |
| O | 8 | K-series | 14.92 | 15.70 | 29.51 | 24.7 |
| | | Total: | 95.08 | 100.00 | 100.00 | |

Spectrum: _1 535

| El | AN | Series | unn. C [wt.-%] | norm. C [wt.-%] | Atom. C [at.-%] | Error [%] |
|---|---|---|---|---|---|---|
| Ti | 22 | K-series | 65.12 | 70.73 | 43.52 | 1.9 |
| N | 7 | K-series | 10.02 | 10.89 | 22.90 | 13.5 |
| C | 6 | K-series | 1.34 | 1.46 | 3.57 | 0.4 |
| Si | 14 | K-series | 1.19 | 1.30 | 1.36 | 0.1 |
| Cl | 17 | K-series | 0.14 | 0.15 | 0.12 | 0.0 |
| O | 8 | K-series | 14.26 | 15.49 | 28.52 | 23.6 |
| | | Total: | 92.07 | 100.00 | 100.00 | |

d)

**Figures:**

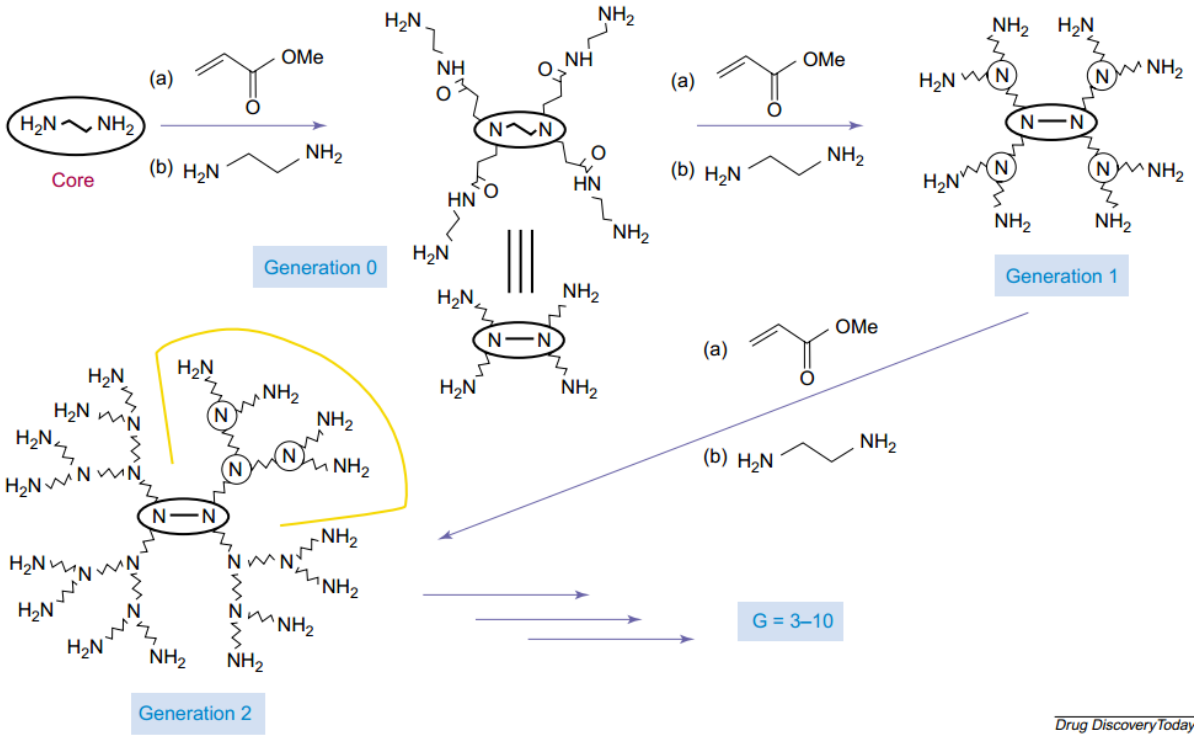



Fig. 1: The synthesis of PAMAM dendrimers begin with an ethylenediamine core with repetitive alkylations with each generation of polymer to generate a fractal molecule. [Esfand et al., 2001]

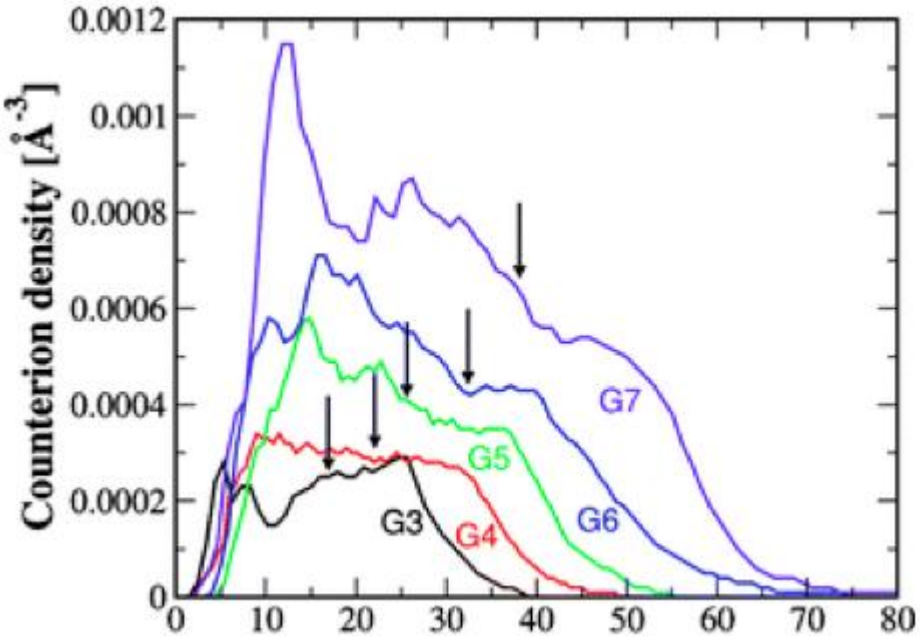



Fig. 2: Chloride counterion distribution in a PAMAM dendrimer, based on molecular dynamics calculations. The arrows indicate the edge of the dendrimer surface. [Maiti et al., 2008]

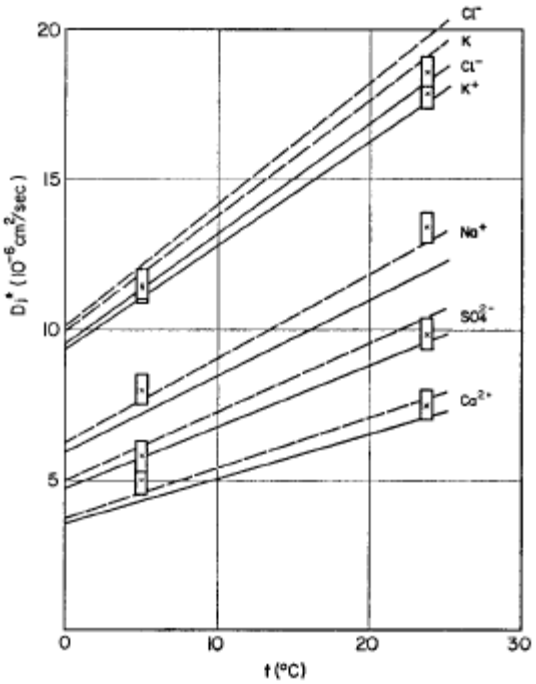



Fig. 3: Chloride ion diffusivity in salt water as a function of temperature [Li and Gregory, 1974].

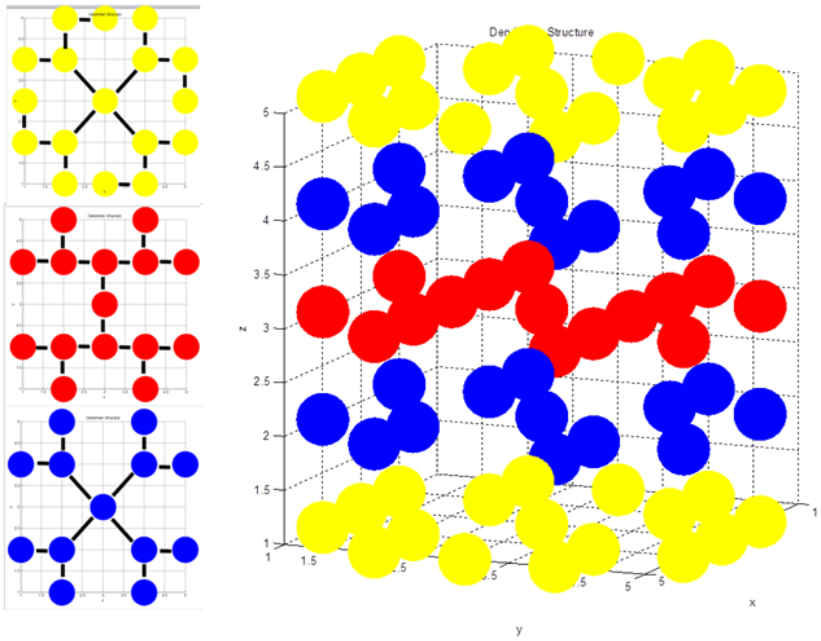


Fig. 4: An illustration of the layers of the approximated dendrimer molecule.

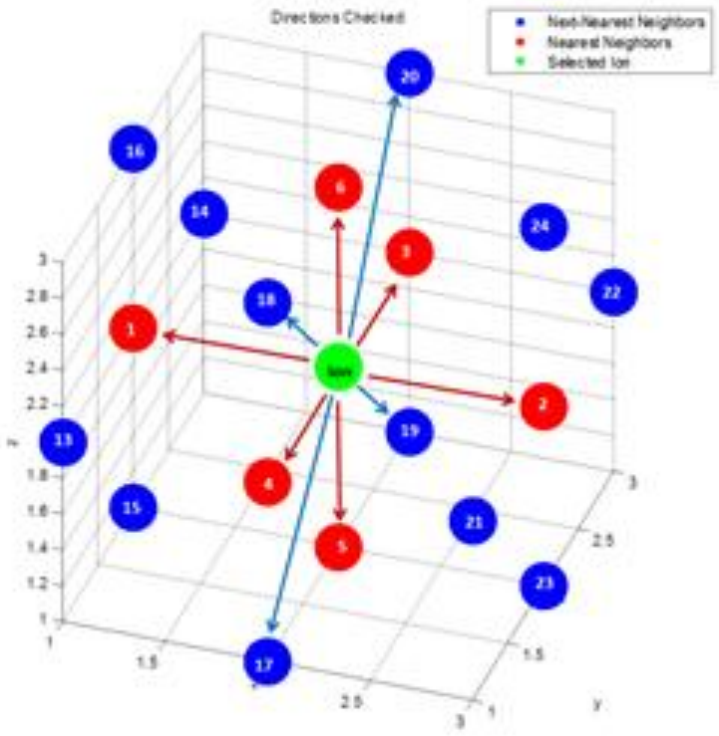



Fig. 5: A schematic of the 18 basic hopping directions considered for each selected ion (illustrated in green). The 6 orthogonal directions (up, down, left, right, in, and out) are illustrated in red, while the twelve diagonal directions are illustrated in blue. The positions shown include all possible single hops, while double hops constitute an additional hop in each direction.

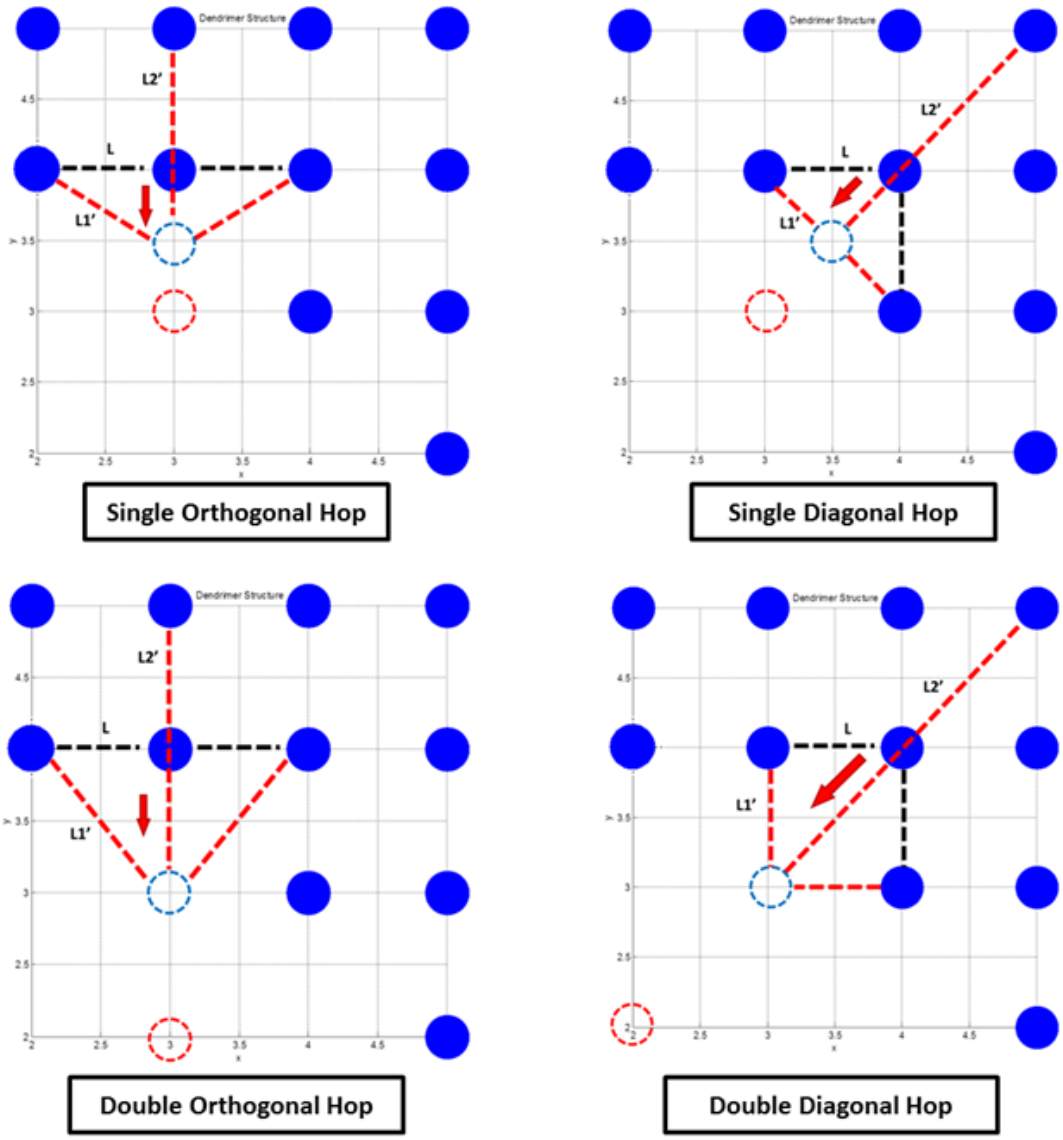



Fig. 6: An illustration of the geometry used to calculate the relative likelihood of single- and double- hops in the orthogonal and diagonal directions.

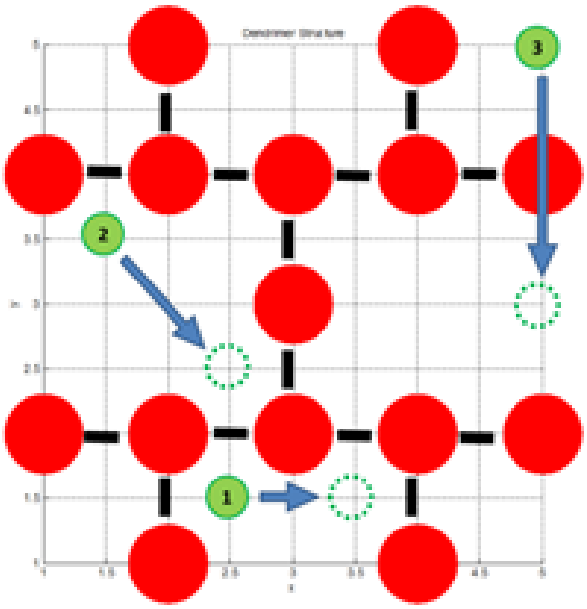



Fig. 7: An illustration of the three basic types of hopping: orthogonally (1), diagonally (2), and across a branch (3).

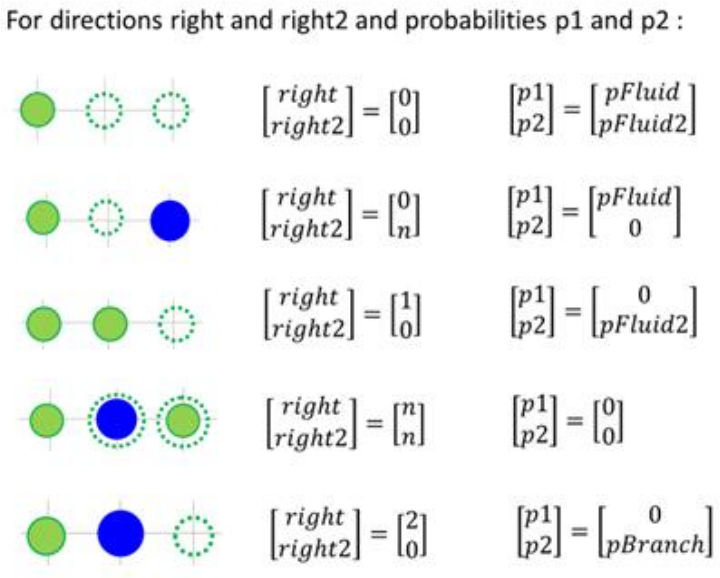
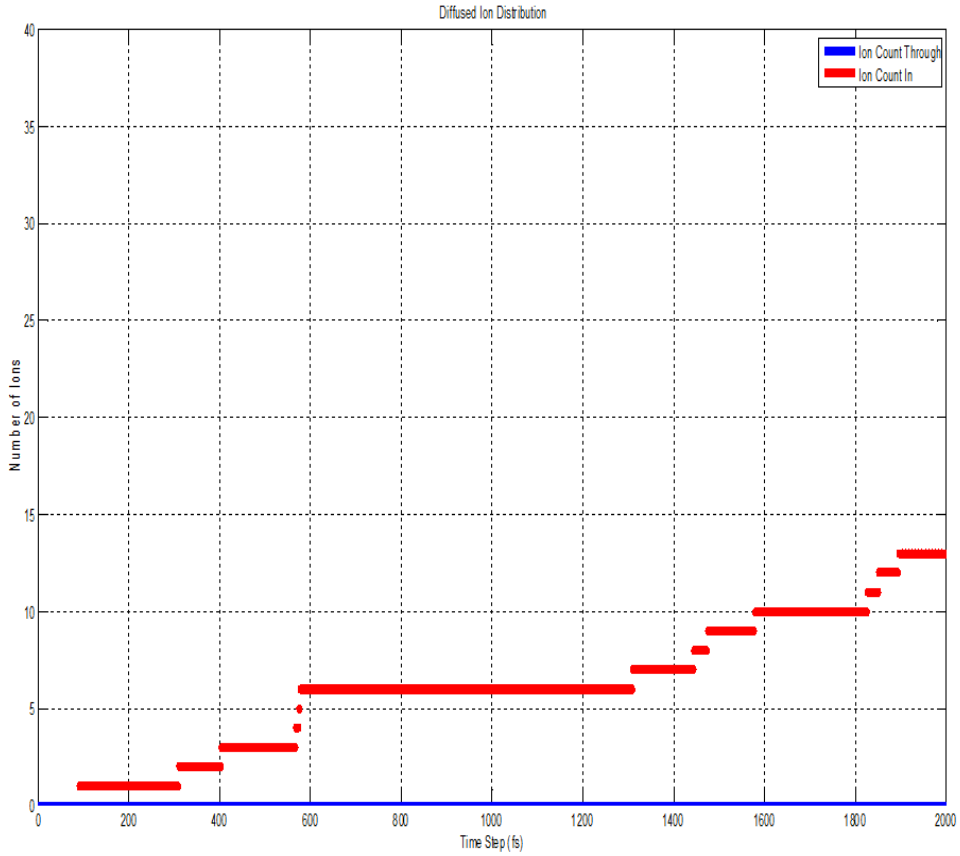
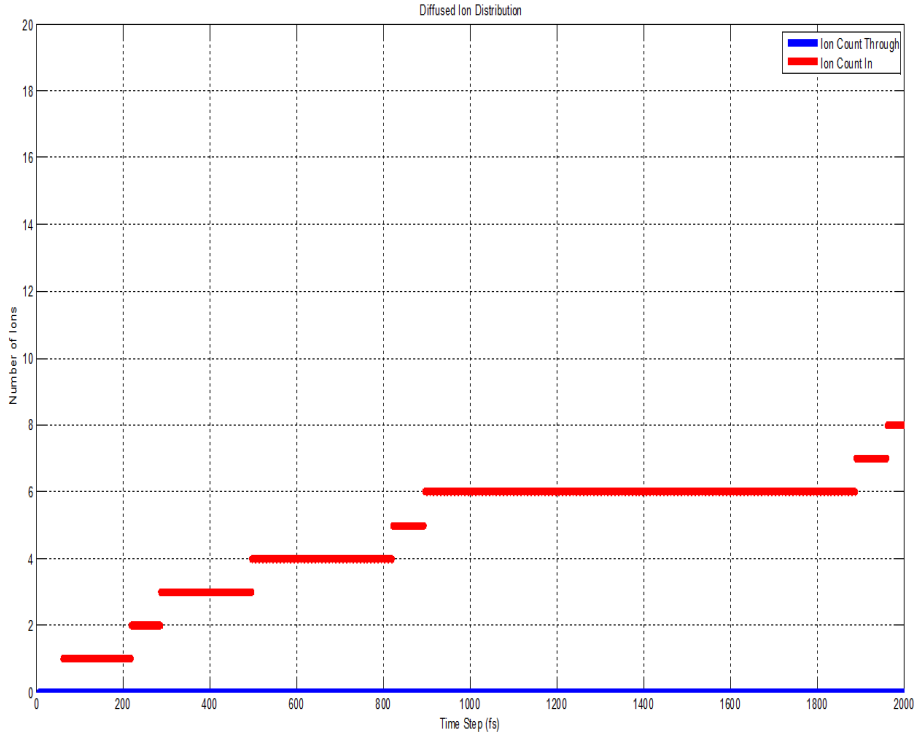
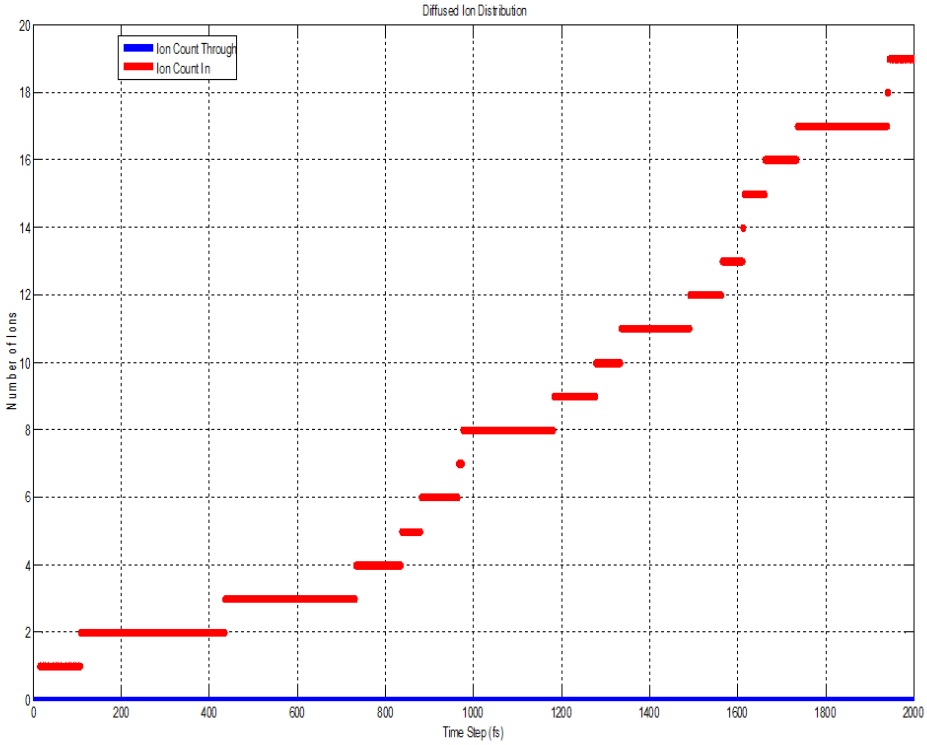



Fig. 8: An illustration of how probabilities are assigned to sets of single and double hops in a given direction. In the example, right and right2 are the single and double hop right, p1 and p2 the corresponding hopping probabilities to each position, pFluid and pFluid2 the single and double hop probabilities in fluid, and pBranch the probability of hopping over a branch. The variable n represents an occupied site (either a 1,ion or 2, dendrimer point). Diagonal directions are similarly defined, with pFluid and pFluid 2 becoming pDiag and pDiag2.


Diffused Ion Distribution
Ion Count Through
Ion Count In
Number of Ions
Time Step (fs)
0
5
10
15
20
25
30
35
40
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Diffused Ion Distribution
Ion Count Through
Ion Count In
Number of Ions
Time Step (fs)
0
2
4
6
8
10
12
14
16
18
20
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Diffused Ion Distribution
Ion Count Through
Ion Count In
Number of Ions
Time Step (fs)
0
2
4
6
8
10
12
14
16
18
20
0
200
400
600
800
1000
1200
1400
1600
1800
2000

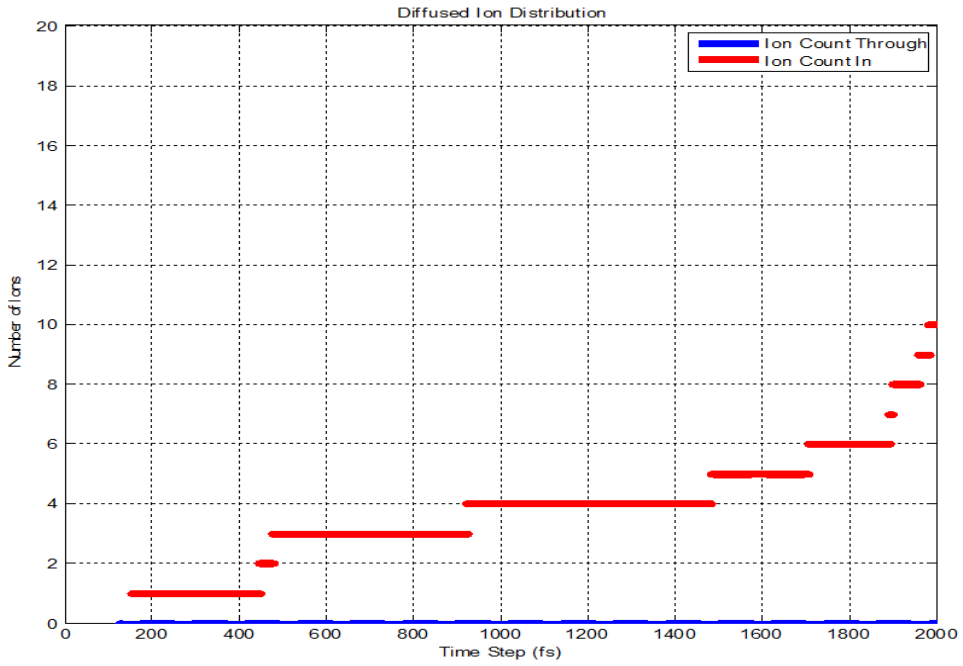



Fig. 9: Sample plots of ion count in the dendrimer film (red) and ion count through the film (blue). No ion diffusion through the film was observed in any trials for the duration of the simulation. All plots shown were resultant of simulations with 2000 time steps, for a simulated area of 24 x 24 nm, and a solution volume was 2765 $nm^3$. The solution was defined to contain 3,000 ions, for a net molarity of 1.80 x $10^{24}$ $M*nm^{-3}$.

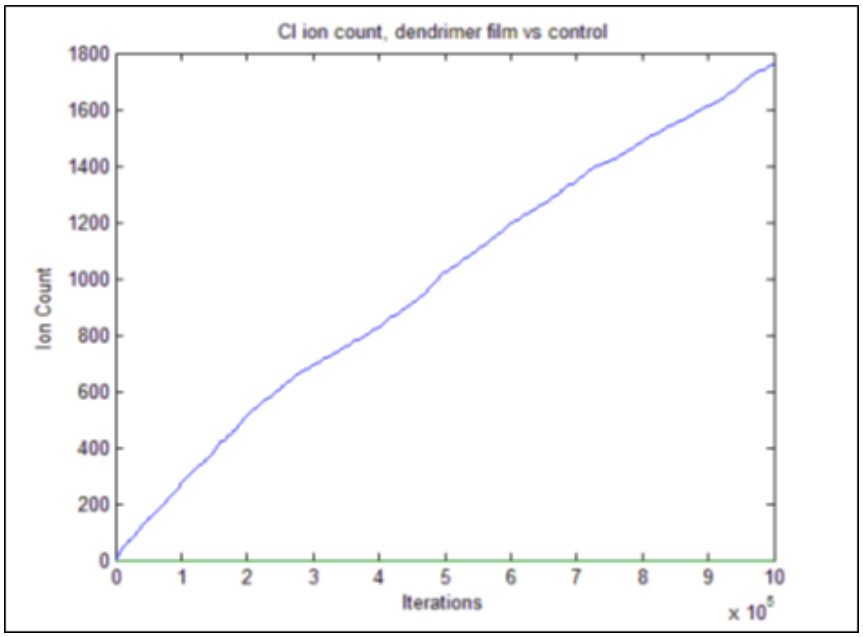



Figure 10: Plot showing ions in $TiO_2$ vs number of iterations, control in blue. Flux was constant through the film

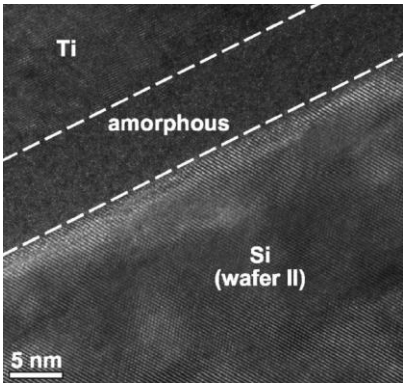
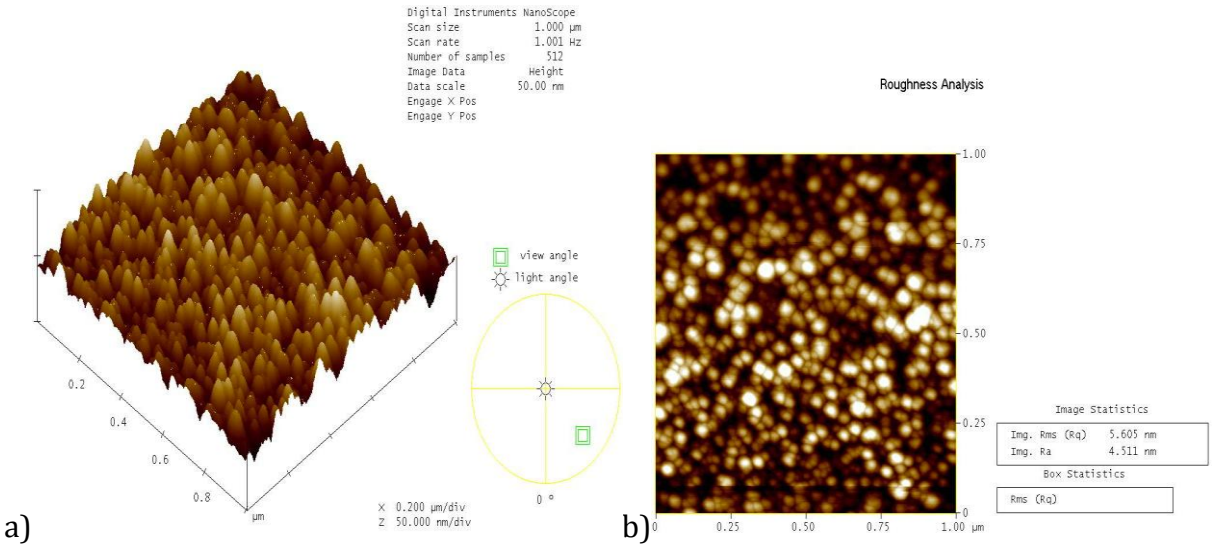
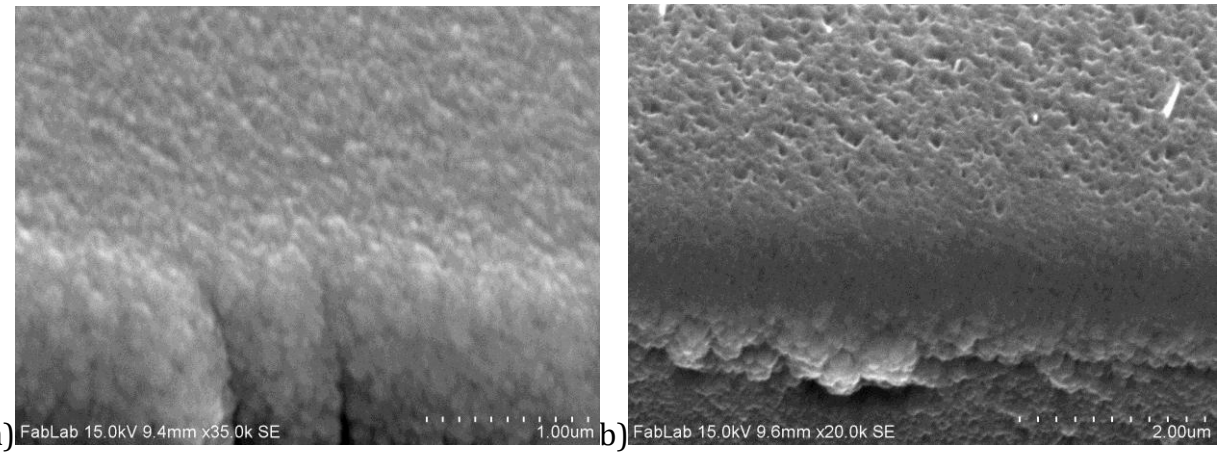



Fig. 11: Transmission electron microscopy (TEM) images reveal an amorphous Si/Ti interface exists between titanium deposited by sputtering and a silicon wafer substrate [Yu et al., 2007].




a) b)

Fig. 12: Sputtered titanium AFM images (a) and RMS roughness calculation (b).




a) b)

Fig. 13: SEM images of the thermally oxidized (a) and plasma oxidized (b) samples at a 45 degree angle from the edge.

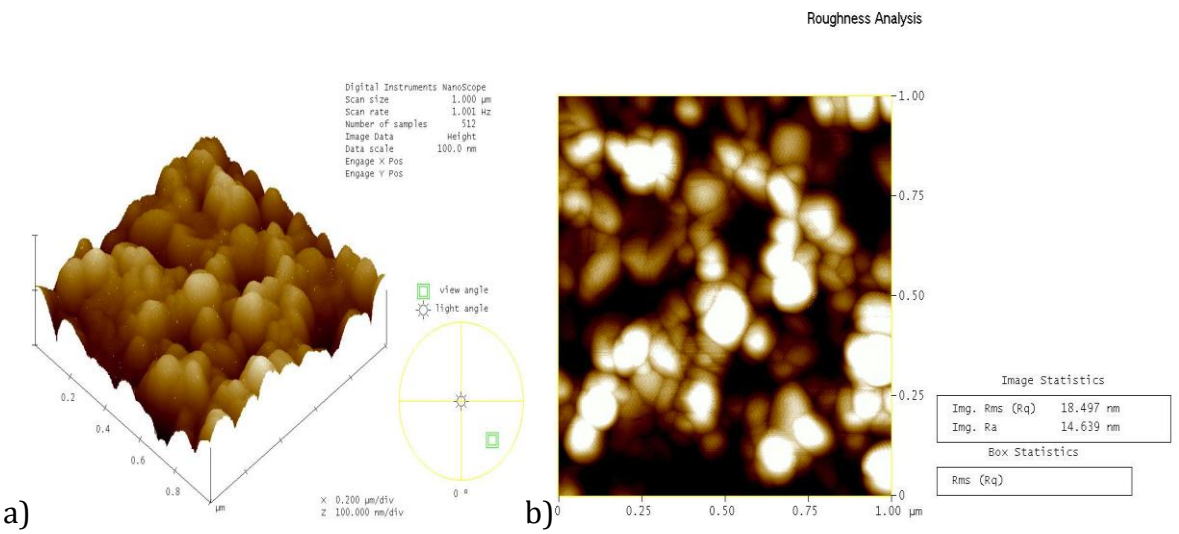



Fig. 14: Thermally oxidized titanium AFM images (a) and RMS roughness calculation (b).

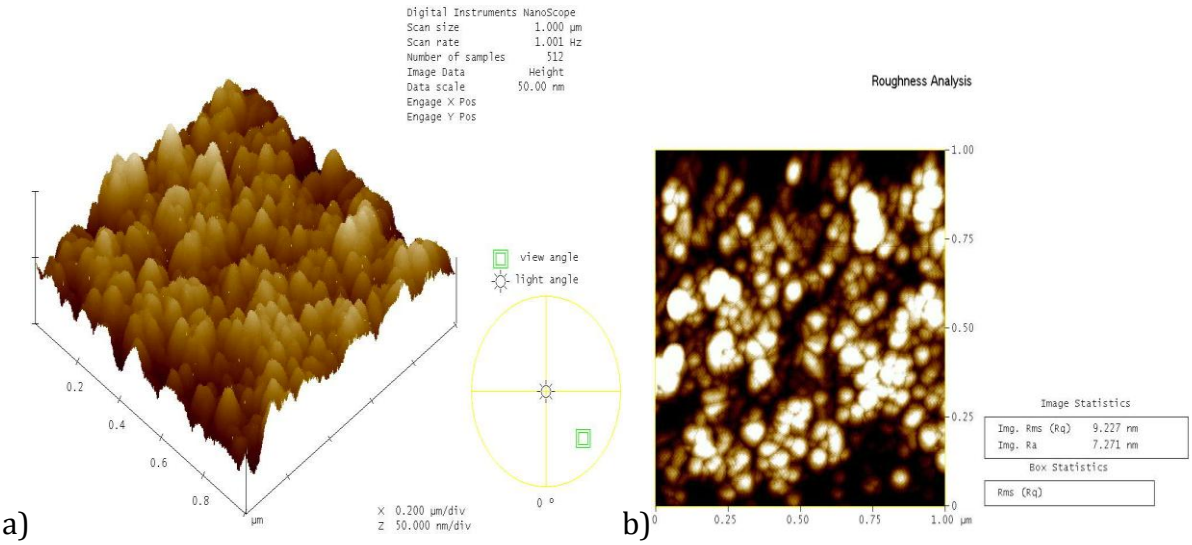
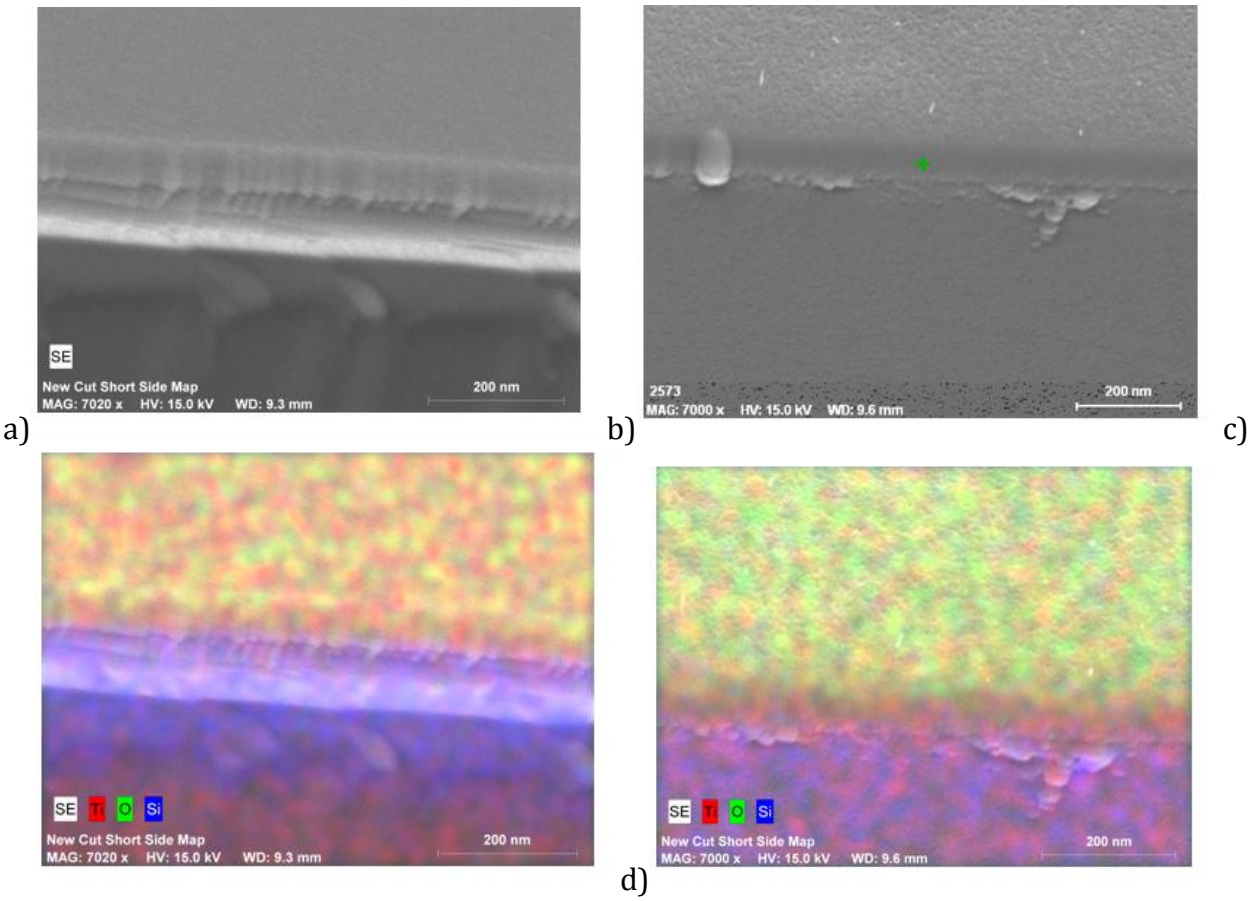



Fig. 15: Plasma oxidized titanium AFM images (a) and RMS roughness calculation (b).




a) b) c) d)

Fig. 16: SEM images and EDS mapping for the thermally oxidized (a,c) and plasma oxidized (b, d) samples. The EDS map demonstrates a fully covered titanium oxide layer on the top of the silicon substrates.

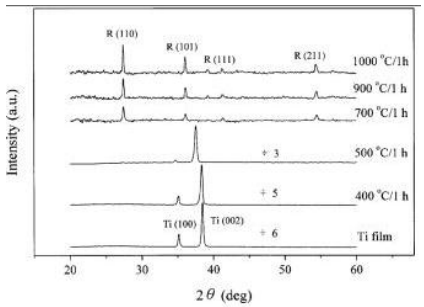



Fig. 17: Based on XRD results, the rutile phase appears when annealing titanium for 1 hour at or above 700°C for thermal oxidation [Ting and Chen, 2000].

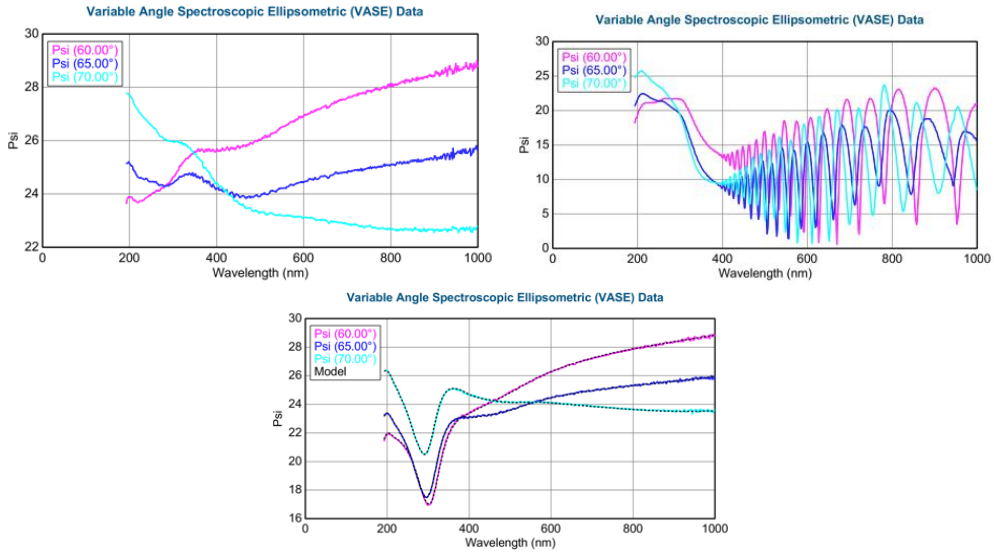



Fig. 18: Ellipsometry data of (a) the titanium substrate, (b) the thermally oxidized titanium, and (c) the plasma oxidized titanium. The fitting of this data gives an oxide thickness of 1600+ nm for the thermally oxidized sample and 1 nm for the plasma oxidized sample.

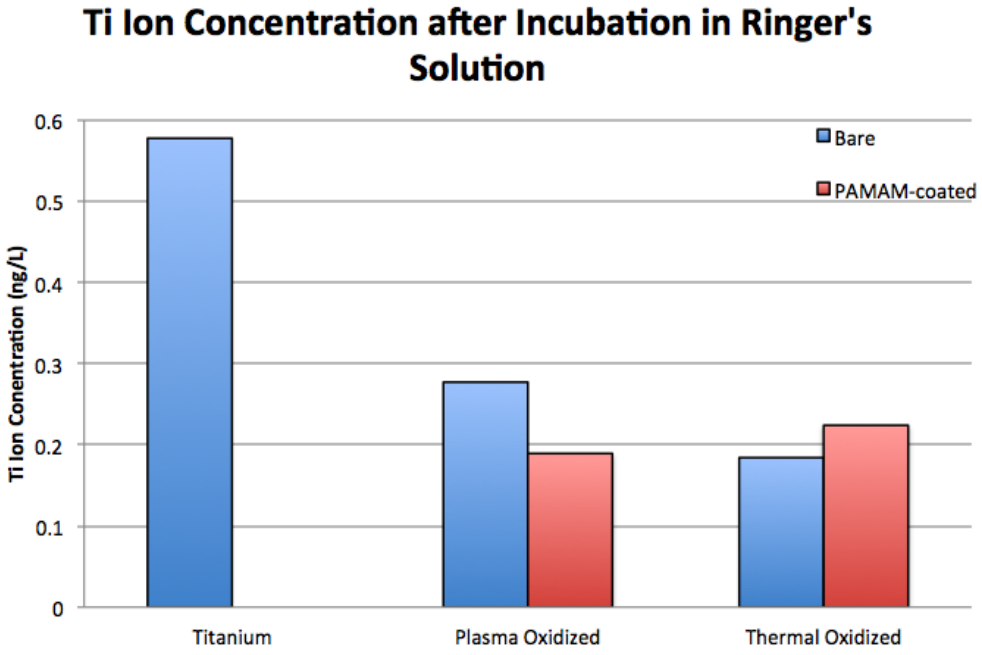



Fig. 19: ICP-OES data after fitting; Calculated concentrations of Ti ions in the Ringer's solution after 5 days of corrosion testing.

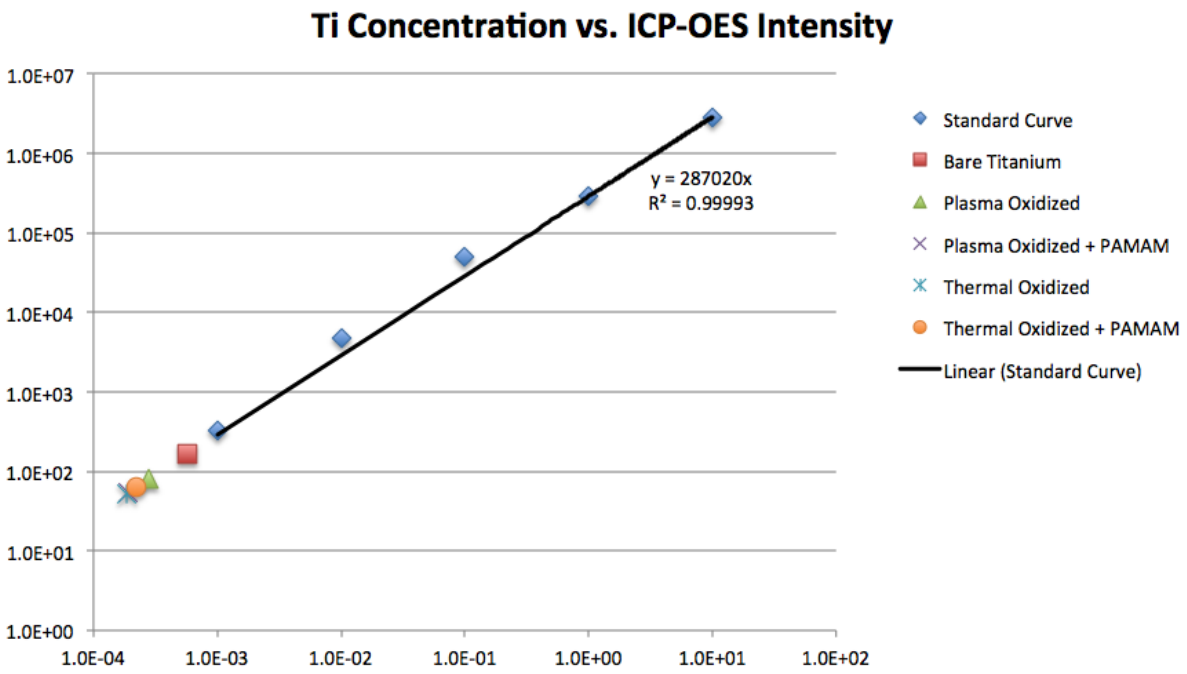



Fig. 20: ICP-OES titanium standard curve and fitting, including the tested samples.

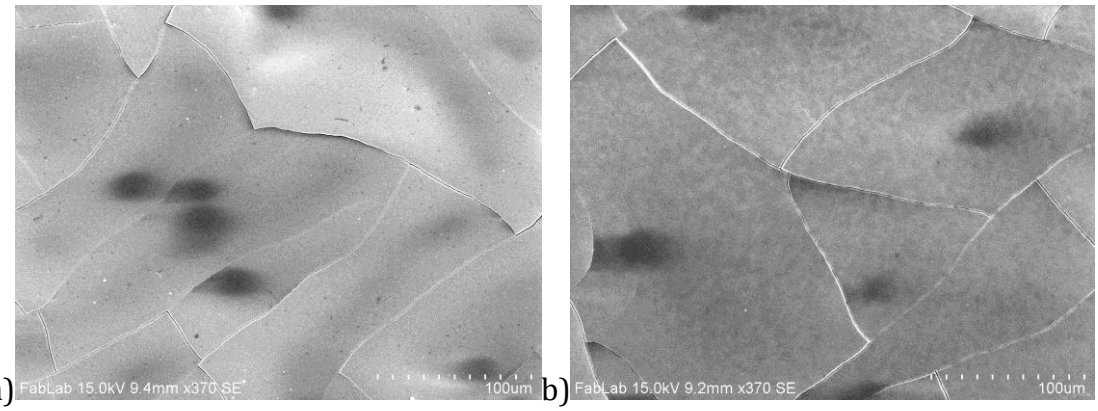



Fig. 21: SEM images of the thermally oxidized samples after incubation for the bare (a) and dendrimer coated (b).

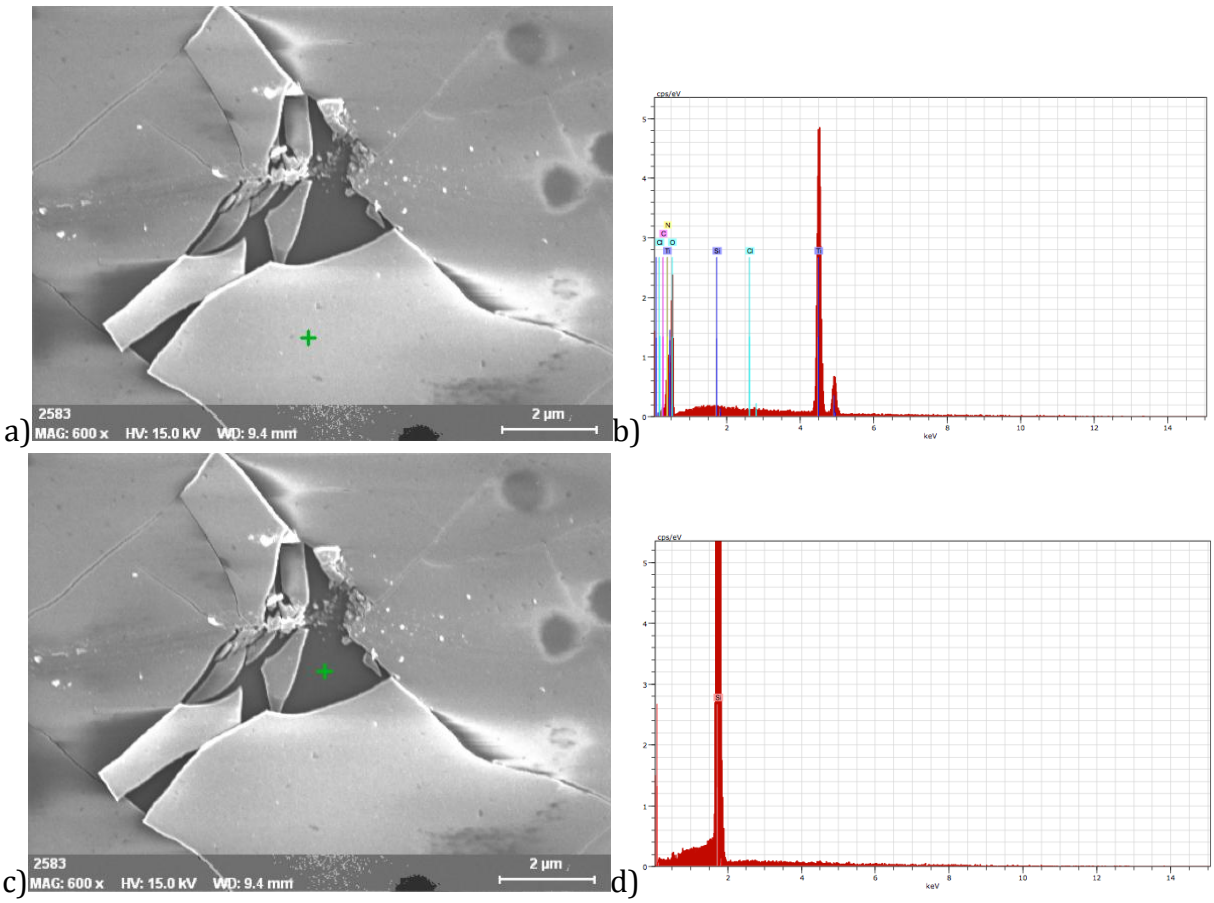



Fig. 22: EDS analysis of openly cracked area of thermally oxidized samples without dendrimer coating reveal the silicon wafer underneath.

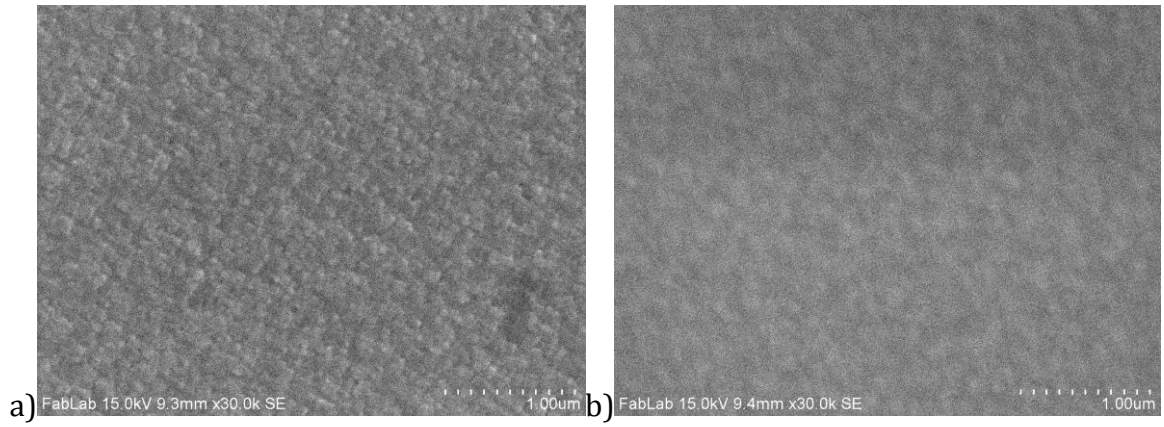



Fig. 23: SEM images of the plasma oxidized surface after incubation for the bare (a) and the dendrimer coated (b) samples.

**Appendix:**

The System Model

Passivation.m:

```
clear()

global perm_of_water charge grid_distance gridxmax gridymax gridzmax boltz temp time_step R
perm_of_water = 80.1*8.854*10^-12;
charge = -1.601*10^-19;
grid_distance = 5*10^-9;
boltz = 1.38*10^-23;
temp = 310;
time_step = 2*10^-15;
R = 5.189*10^19;
num_iterations = 1000000;


gridxmax = 10; gridymax = 10; gridzmax = 20;
volume = zeros(gridxmax, gridymax, gridzmax);

volume(:, :, 1) = 3.*ones(gridxmax, gridymax, 1);
ti_coords = zeros(gridxmax*gridymax, 3);
for i = 1:gridxmax*gridymax
   [x, y, z] = ind2sub([gridxmax gridymax 1], i);
   ti_coords(i, 1) = x; ti_coords(i, 2) = y; ti_coords(i, 3) = 1;
end

cl_per_cubic_nanometer = .07;
system_volume = (gridxmax)*(gridxmax)*((gridzmax - 1));
number_cl_ions = floor(cl_per_cubic_nanometer*system_volume);
cl_initial_locations = randperm(gridxmax*gridymax*(gridzmax-1), number_cl_ions);
for i = 1:length(cl_initial_locations)
   index = cl_initial_locations(i);
   [x, y, z] = ind2sub([gridxmax, gridymax, gridzmax-1], index);
   volume(x, y, z+1) = 1;
end
```

```
num_diffused_cl_in_ti = 0;

tracking = zeros(num_iterations, 1);

for n = 1:num_iterations
  if mod(log10(n), 1) == 0
    fprintf('Beginning iteration %f\n', n);
  end

  cl_locations = find(volume == 1);
  ion_choice = cl_locations(randi(length(cl_locations)));
  [x, y, z] = ind2sub([gridxmax, gridymax, gridzmax], ion_choice);
  direction = getHoppingProbabilities([x y z], volume);
  volume(ion_choice) = 0;

  if x+direction(1) > 0 && x+direction(1) < gridxmax && y+direction(2) > 0 &&
y+direction(2) < gridymax && z + direction(3) > 0 && z + direction(3) < gridzmax
    endLocation = sub2ind([gridxmax, gridymax, gridzmax], x+direction(1), y+direction(2),
z+direction(3));
  else
    endLocation = ion_choice;
  end

  if volume(endLocation) == 3
    num_diffused_cl_in_ti = num_diffused_cl_in_ti + 1;
    fprintf('Chlorine ion %f has entered the bulk\n', num_diffused_cl_in_ti)
  else
    volume(endLocation) = 1;
  end
  tracking(n, 1) = num_diffused_cl_in_ti;

  if length(cl_locations) < number_cl_ions
%     disp('Injecting chlorine ion into solution')
    open_locations = find(volume == 0);
    rand_index = randi(length(open_locations), 1);
    volume(open_locations(rand_index)) = 1;
  end

  plot(tracking(1:n, 1));
  drawnow
```

```
%    cl_coords = zeros(length(cl_locations), 3);
%    for i = 1:length(cl_locations)
%       [x, y, z] = ind2sub([gridxmax gridymax gridzmax], cl_locations(i));
%       cl_coords(i, 1) = x; cl_coords(i, 2) = y; cl_coords(i, 3) = z;
%    end
%    scatter3(ti_coords(:, 1), ti_coords(:, 2), ti_coords(:, 3), 'k', 's', 'fill')
%    hold on
%    grid on
%    scatter3(cl_coords(:, 1), cl_coords(:, 2), cl_coords(:, 3), 'r')
% %    scatter3(dendrimer_coords(:, 1), dendrimer_coords(:, 2), dendrimer_coords(:, 3), 'm',
'fill')
%    view(-30, 10)
%    hold off
%    drawnow
end
```

getHoppingProbabilities.m

```
function [direction] = getHoppingProbabilities(location, matrix)

global gridxmax gridymax gridzmax time_step R temp

x = location(1);
y = location(2);
z = location(3);
possible_directions = [-1 0 0; 1 0 0; 0 -1 0; 0 1 0; 0 0 -1; 0 0 1];
xmin = max(1, x-2); xmax = min(gridxmax, x+2);
ymin = max(1, y-2); ymax = min(gridymax, y+2);
zmin = max(1, z-2); zmax = min(gridzmax, z+2);

probabilities = zeros(1,6);

probabilities(1) = jumpProbability(matrix, [max(x-1, xmin) y z], xmin, x, ymin+1, ymax-1,
zmin+1, zmax-1);
probabilities(2) = jumpProbability(matrix, [min(x+1, xmax) y z], x, xmax, ymin+1, ymax-1,
zmin+1, zmax-1);
probabilities(3) = jumpProbability(matrix, [x max(y-1, ymin) z], xmin+1, xmax-1, ymin, y,
zmin+1, zmax-1);
probabilities(4) = jumpProbability(matrix, [x min(y+1, ymax) z], xmin+1, xmax-1, y, ymax,
zmin+1, zmax-1);
```

```
probabilities(5) = jumpProbability(matrix, [x y max(z-1,zmin)], xmin+1, xmax-1, ymin+1, ymax-1, zmin, z);
probabilities(6) = jumpProbability(matrix, [x y min(z+1,zmax)], xmin+1, xmax-1, ymin+1, ymax-1, z, zmax);

if x == 1 || matrix(x-1, y, z) == 1
   probabilities(1) = Inf;
end
if x == gridxmax || matrix(x+1, y, z) == 1
   probabilities(2) = Inf;
end
if y == 1 || matrix(x, y-1, z) == 1
   probabilities(3) = Inf;
end
if y == gridymax || matrix(x, y+1, z) == 1
   probabilities(4) = Inf;
end
if z == 1 || matrix(x, y, z-1) == 1
   probabilities(5) = Inf;
end
if z == 2
   probabilities(1) = probabilities(1) + (.76/(R*temp));
   probabilities(2) = probabilities(2) + (.76/(R*temp));
   probabilities(3) = probabilities(3) + (.76/(R*temp));
   probabilities(4) = probabilities(4) + (.76/(R*temp));
   probabilities(5) = probabilities(5) + (2.89/(R*temp));
end
if z == 3
   probabilities(5) = probabilities(5) + (-2.89/(R*temp));
end
if z == gridzmax || matrix(x, y, z+1) == 1
   probabilities(6) = Inf;
end

probabilities = time_step * exp(-1*probabilities);
totalProb = sum(probabilities);
numNeighborsMod = probabilities./totalProb;

for i = 6:-1:2
   numNeighborsMod(i) = sum(numNeighborsMod(1:i));
```

```
end

if isequal([0 0 0 0 0 0], probabilities)
   chosen_index = randi(6);
else
   r = rand(1);
   chosen_index = find(numNeighborsMod > r, 1);
end

direction = possible_directions(chosen_index, :);


end
```

jumpProbability.m

```
function [deltaG] = jumpProbability(matrix, focus, xmin, xmax, ymin, ymax, zmin, zmax)

global charge perm_of_water grid_distance gridxmax gridymax gridzmax temp boltz
deltaG = 0;

if matrix(sub2ind([gridxmax gridymax gridzmax], focus(1), focus(2), focus(3))) ~= 1   &&
matrix(sub2ind([gridxmax gridymax gridzmax], focus(1), focus(2), focus(3))) ~= 2
   for i = xmin:xmax
      for j = ymin:ymax
         for k = zmin:zmax
            probe = matrix(sub2ind([gridxmax gridymax gridzmax], i, j, k));
            if ~isequal(focus, [i j k])
               if probe == 1
                  dist = ([i j k] - focus).*grid_distance;
                  dist = dist.^2;
                  deltaG                    =                    deltaG                    +
((charge)^2/(4*pi()*sum(dist)^.5*perm_of_water*grid_distance)/(boltz*temp));
               end
%                 if probe == 2
%                    location = sub2ind([gridxmax gridymax], i, j);
%                    dist = [i j k] - focus;
%                    dist = dist.^2;
```

```
%                                                    deltaG = deltaG - (64-
dendrimer_charge(location))*charge^2/(4*pi()*sum(dist)^.5*perm_of_water*grid_distanc
e) + boltz*temp*(dendrimer_charge(location)-1);
%               end
%               if probe == 3
%                  deltaG = deltaG + (.58);
%               end
          end
       end
    end
  end
end

deltaG = deltaG/(temp*boltz);

end
```

Dendrimer Model

KMC Simulation of Cl- Ion Diffusion Through a Dendrimer Film

by Elizabeth Ashley



*Definition of global variables*

For the constraining volume and solution:

```
global x y z Particles stepmax  % x, y, and z must be multiples of 5 (or however big the
dendrimer unit cell is)
x=40; y=40; z=20; Particles=3000; stepmax = 1;
% For the dendrimer:
global a b c D15 D24 D3
a=5; b=5; c=5;  % Defining the size of the dendrimer "unit cell"
```

*Definition and Visualization: Dendrimer and Cl- ion solution*

```
D15 = [0 2 2 2 0;2 2 0 2 2;2 0 2 0 2;2 2 0 2 2;0 2 2 2 0];
D24 = [0 2 0 2 0;2 2 0 2 2;0 0 2 0 0;2 2 0 2 2;0 2 0 2 0]; % The dendrimer atoms are defined as
2s for now
D3 = [0 2 0 2 0;2 2 0 2 2;0 2 2 2 0;2 2 0 2 2;0 2 0 2 0];

% To create the dendrimer unit cell- approximated as cubic
Dendrimer1=zeros(a,b,c);
    Dendrimer1(:,:,1)= D15;
    Dendrimer1(:,:,5)= D15;

Dendrimer2=zeros(a,b,c);
    Dendrimer2(:,:,2)= D24;
    Dendrimer2(:,:,4) = D24;

Dendrimer3=zeros(a,b,c);
    Dendrimer3(:,:,3) = D3;

FindDen1=find(Dendrimer1);
[Iden1,Jden1,Kden1]=ind2sub(size(Dendrimer1),FindDen1);

FindDen2=find(Dendrimer2);
[Iden2,Jden2,Kden2]=ind2sub(size(Dendrimer2),FindDen2);

FindDen3=find(Dendrimer3);
[Iden3,Jden3,Kden3]=ind2sub(size(Dendrimer3),FindDen3);

% This plots the dendrimer molecule
% clf
% hold on
% figure(1)
% scatter3(Iden1,Jden1,Kden1,2000,'y','filled');
% scatter3(Iden2,Jden2,Kden2,2000,'b','filled');
% scatter3(Iden3,Jden3,Kden3,2000,'r','filled');
% title('Dendrimer Structure');
% xlabel('x');
% ylabel('y');
% zlabel('z');
% grid on
% hold off

% To create the tiled dendrimer "film" layer:
Dendrimer=zeros(a,b,c);
UC15=[D15 D15];
UC24=[D24 D24];
UC3=[D3 D3];
    Dendrimer(:,:,1)=D15;
    Dendrimer(:,:,5)=D15;
    Dendrimer(:,:,2)=D24;
    Dendrimer(:,:,4)=D24;
    Dendrimer(:,:,3)=D3;
```

```
 DenLatt=repmat(Dendrimer,x/5,y/5);
 FindDenLatt=find(DenLatt);
 [IDenLatt,JDenLatt,KDenLatt]=ind2sub(size(DenLatt),FindDenLatt);
% figure(2) % To plot the dendrimer lattice
% scatter3(IDenLatt,JDenLatt,KDenLatt,30,'b','filled');
% title('Tiled Dendrimer Lattice');
% xlabel('x');
% ylabel('y');
% zlabel('z');
```

*Creating the Initial System: Constraining volume, dendrimer film, and Cl- ion solution*

```
VolMat= zeros(x,y,z);  % Defines the size of the initial constraining volume
VolMat(:,:,(12))= DenLatt(:,:,1); % This creates the dendrimer layers
VolMat(:,:,(11))= DenLatt(:,:,2);
VolMat(:,:,(10))= DenLatt(:,:,3);
VolMat(:,:,(9))= DenLatt(:,:,4);
VolMat(:,:,(8))= DenLatt(:,:,5);

% Creating the random Cl- ion solution: ions are defined as 1s
SolMat=ones(x,y,(z-12));
Indices=[1:numel(SolMat)]';
[I,J,K]= ind2sub(size(SolMat),Indices);
OneVolume = find(SolMat == 1);              % Finds the indices of all entries containing ones
ChosenOnes=randsample(OneVolume,Particles); % Defines the concentration of particles in the
matrix
[I1,J1,K1]=ind2sub(size(SolMat),ChosenOnes);
Aind=[I J K];
Bind=[I1 J1 K1];
CommonOnes=ismember(Aind,Bind,'rows');      % Inserts 0s in all locations not chosen to be an ion
    for n = 1:length(CommonOnes);
        if CommonOnes(n)== 0;
        SolMat(n)=0;
        end
    end
 VolMat(:,:,(13:z))=SolMat;                 % Inserts the solution volume into the matrix for the
combined system
% Additional constraints:
VolMat(1,:,:)=0;                            % Defining boundary conditions of the entire system
VolMat(2,:,:)=0;
VolMat(x,:,:)=0;
VolMat(x-1,:,:)=0;
VolMat(:,1,:)=0;
VolMat(:,2,:)=0;
VolMat(:,y,:)=0;
VolMat(:,y-1,:)=0;
VolMat(:,:,1)=0;
VolMat(:,:,2)=0;
VolMat(:,:,z)=0;
VolMat(:,:,z-1)=0;
```

*Visualization of initial particle distribution*

This section generates a 3D plot of the dendrimer film and initial Cl- particle distribution (step==1)

```
% VolMat2=VolMat;
% figure(3);
% FindVmDen=find(VolMat2==2);
%     [IVmDen JVmDen KVmDen]=ind2sub(size(VolMat2),FindVmDen);
% FindVmIon=find(VolMat2==1);
%     [IVmIon JVmIon KVmIon]=ind2sub(size(VolMat2),FindVmIon);
%
% hold on;
% scatter3(IVmIon,JVmIon,KVmIon,30,'g','filled');
% scatter3(IVmDen,JVmDen,KVmDen,200,'r','filled');
%
% title('Initial State of Combined System');
% xlabel('x');
% ylabel('y');
% zlabel('z');
% hold off
```

*Choosing a random particle (ion) to move*

```
VolMat2=VolMat;                         % VolMat2 is a 4D matrix that stores the 3D positions of
all elements in VolMat per time step
for step=1:stepmax;                     % Time steps are intervals of femtoseconds (set by hopping
rates)
VolMat2(:,:,:,step)=VolMat;
OnesVm = find(VolMat2(:,:,:,step)==1);
tot=0; AA=0; CHECK=0;
A=size(OnesVm,1);
while tot~=1 && AA~=1 && CHECK~=1;      % Ensures that movement of a chosen ion only proceeds if it
is within the allowed boundary
OneSum = cumsum(OnesVm);
    if OneSum~=0
        CHECK=1;
    elseif OneSum==0;
        CHECK=0;                        % CHECK fixes a random error that occasionally causes the
sum of all ions (1s) in the system to be read as 0.
    end
r1=rand(1);
OneFinder = r1*OneSum(end);
OneChoose=find(OneSum >= OneFinder,1,'first');
AtomChoose=OnesVm(OneChoose);
% Begin selection and movement of a particle
    [I,J,K]=ind2sub(size(VolMat2(:,:,:)),AtomChoose); % ind2sub: converts the indices of points
to [I J K] subscripts
    OnesVm(OneChoose)=[];
    A=size(OnesVm,1);
    if  I > 2 && I < x-1 && J > 2 && J < y-1 && K > 2 && K < z-1 % Ensures an ion cannot move
beyond the boundary of the constraining volume
        tot=tot+1;
        AA=1;
    Position=struct();    % saves the position of the chosen ion in a structure file for easy
```

```
reference
        Position.xi=I;
        Position.yi=J;
        Position.zi=K;

%  Definition of directions for movement (36 total)
Direc=struct();
    % Nearest neighbors (6 in SC lattice)
    Direc.up=VolMat2(Position.xi-1,Position.yi,Position.zi);
    Direc.down=VolMat2(Position.xi+1,Position.yi,Position.zi);
    Direc.right=VolMat2(Position.xi,Position.yi+1,Position.zi);
    Direc.left=VolMat2(Position.xi,Position.yi-1,Position.zi);
    Direc.out=VolMat2(Position.xi,Position.yi,Position.zi-1);
    Direc.in=VolMat2(Position.xi,Position.yi,Position.zi+1);
    % Second nearest neighbors
    Direc.up2=VolMat2(Position.xi-2,Position.yi,Position.zi);
    Direc.down2=VolMat2(Position.xi+2,Position.yi,Position.zi);
    Direc.right2=VolMat2(Position.xi,Position.yi+2,Position.zi);
    Direc.left2=VolMat2(Position.xi,Position.yi-2,Position.zi);
    Direc.out2=VolMat2(Position.xi,Position.yi,Position.zi-2);
    Direc.in2=VolMat2(Position.xi,Position.yi,Position.zi+2);
    % First diagonal neighbors
    Direc.upleft=VolMat2(Position.xi-1,Position.yi-1,Position.zi);
    Direc.upright=VolMat2(Position.xi-1,Position.yi+1,Position.zi);
    Direc.upout=VolMat2(Position.xi-1,Position.yi,Position.zi-1);
    Direc.upin=VolMat2(Position.xi-1,Position.yi,Position.zi+1);
    Direc.leftout=VolMat2(Position.xi,Position.yi-1,Position.zi-1);
    Direc.leftin=VolMat2(Position.xi,Position.yi-1,Position.zi+1);
    Direc.rightout=VolMat2(Position.xi,Position.yi+1,Position.zi-1);
    Direc.rightin=VolMat2(Position.xi,Position.yi+1,Position.zi+1);
    Direc.downleft=VolMat2(Position.xi+1,Position.yi-1,Position.zi);
    Direc.downright=VolMat2(Position.xi+1,Position.yi+1,Position.zi);
    Direc.downout=VolMat2(Position.xi+1,Position.yi,Position.zi-1)
    Direc.downin=VolMat2(Position.xi+1,Position.yi,Position.zi+1);
    % Second diagonal neighbors
    Direc.upleft2=VolMat2(Position.xi-2,Position.yi-2,Position.zi);
    Direc.upright2=VolMat2(Position.xi-2,Position.yi+2,Position.zi);
    Direc.upout2=VolMat2(Position.xi-2,Position.yi,Position.zi-2);
    Direc.upin2=VolMat2(Position.xi-2,Position.yi,Position.zi+2);
    Direc.leftout2=VolMat2(Position.xi,Position.yi-2,Position.zi-2);
    Direc.leftin2=VolMat2(Position.xi,Position.yi-2,Position.zi+2);
    Direc.rightout2=VolMat2(Position.xi,Position.yi+2,Position.zi-2);
    Direc.rightin2=VolMat2(Position.xi,Position.yi+2,Position.zi+2);
    Direc.downleft2=VolMat2(Position.xi+2,Position.yi-2,Position.zi);
    Direc.downright2=VolMat2(Position.xi+2,Position.yi+2,Position.zi);
    Direc.downout2=VolMat2(Position.xi+2,Position.yi,Position.zi-2);
    Direc.downin2=VolMat2(Position.xi+2,Position.yi,Position.zi+2);

        DirecCell=struct2cell(Direc);          % Checks the values of VolMat2 at the defined
possible directions and stores them in a structure file
        DirecMat=cell2mat(DirecCell);
            up=DirecMat(1);
            down=DirecMat(2);
```

```
            right=DirecMat(3);
            left=DirecMat(4);
            out=DirecMat(5);
            in=DirecMat(6);

            up2=DirecMat(7);
            down2=DirecMat(8);
            right2=DirecMat(9);
            left2=DirecMat(10);
            out2=DirecMat(11);
            in2=DirecMat(12);

            upleft=DirecMat(13);
            upright=DirecMat(14);
            upout=DirecMat(15);
            upin=DirecMat(16);
            leftout=DirecMat(17);
            leftin=DirecMat(18);
            rightout=DirecMat(19);
            rightin=DirecMat(20);
            downleft=DirecMat(21);
            downright=DirecMat(22);
            downout=DirecMat(23);
            downin=DirecMat(24);

            upleft2=DirecMat(25);
            upright2=DirecMat(26);
            upout2=DirecMat(27);
            upin2=DirecMat(28);
            leftout2=DirecMat(29);
            leftin2=DirecMat(30);
            rightout2=DirecMat(31);
            rightin2=DirecMat(32);
            downleft2=DirecMat(33);
            downright2=DirecMat(34);
            downout2=DirecMat(35);
            downin2=DirecMat(36);
```

*Defining hopping probability in different directions and for various distances*

```
pFluid = .04697/.04697;              % All probabilities are defined relative to fluid hopping,
which is set as 1
pFluid2 = ((1/5.509)*pFluid);        % There is a lower probability for two hops in an
orthogonal direction
pBranch = 1/.04697;                  % There is a higher probability of hopping across a branch
inside the molecule
pFluidDiag=pFluid*(1/3.098);         % Much lower probabilities for both types of diagonal hop
pFluidDiag2=pFluid*(1/13.49);

% To account for increased likelihood of branch hopping inside the
% dendrimer (this approximates the hopping rate increasing with proximity to dendrimer center)
if Position.zi==8 || Position.zi==12;
    IntRate=.0221/.04697;
elseif Position.zi==9 || Position.zi==11;
```

```
    IntRate==1/.04697;
elseif Position.zi == 10;
    IntRate==1/.04697;
else
    IntRate = 1;
end
% For basic hopping to nn open site
if up == 0 && up2 == 0;            % The case where both consecutive sites in the same direction
are open
    p1 = pFluid; p7 = pFluid2;
elseif up ==0 && up2 ~= 0;         % Where the nearest site is open and the second is occupied
    p1 = pFluid; p7 = 0;
elseif up == 1 && up2 == 0;        % Where the nearest site is occupied but the second nearest
site is open
    p1 = 0; p7 = pFluid2;
elseif up ~= 0 && up2 ~= 0;        % Where both consecutive sites are occupied
    p1 = 0; p7 = 0;
elseif up == 2 && up2 == 0;        % Where a branch occupies the nearest site and the consecutive
site is open
    p1 = 0; p7 = pBranch*IntRate;
end

if  down == 0 && down2 == 0;
    p2 = pFluid; p8 = pFluid2;
elseif down ==0 && down2 ~= 0;
    p2 = pFluid; p8 = 0;
elseif down == 1 && down2 == 0;
    p2 = 0; p8 = pFluid2;
elseif down ~= 0 && down2 ~= 0;
    p2 = 0; p8 = 0;
elseif down == 2 && down2 == 0;
    p2 = 0; p8 = pBranch*IntRate;
end

if  right == 0 && right2 == 0;
    p3 = pFluid; p9 = pFluid2;
elseif right ==0 && right2 ~= 0;
    p3 = pFluid; p9 = 0;
elseif right == 1 && right2 == 0;
    p3 = 0; p9 = pFluid2;
elseif right ~= 0 && right2 ~= 0;
    p3 = 0; p9 = 0;
elseif right == 2 && right2 == 0;
    p3 = 0; p9 = pBranch*IntRate;
end

if  left == 0 && left2 == 0;
    p4 = pFluid; p10 = pFluid2;
elseif left ==0 && left2 ~= 0;
    p4 = pFluid; p10 = 0;
elseif left == 1 && left2 == 0;
    p4 = 0; p10 = pFluid2;
elseif left ~= 0 && left2 ~= 0;
    p4 = 0; p10 = 0;
```

```
elseif left == 2 && left2 == 0;
    p4 = 0; p10 = pBranch*IntRate;
end

if  out == 0 && out2 == 0;
    p5 = pFluid; p11 = pFluid2;
elseif out ==0 && out2 ~= 0;
    p5 = pFluid; p11 = 0;
elseif out == 1 && out2 == 0;
    p5 = 0; p11 = pFluid2;
elseif out ~= 0 && out2 ~= 0;
    p5 = 0; p11 = 0;
elseif out == 2 && out2 == 0;
    p5 = 0; p11 = pBranch*IntRate;
end

if  in == 0 && in2 == 0;
    p6 = pFluid; p12 = pFluid2;
elseif in ==0 && in2 ~= 0;
    p6 = pFluid; p12 = 0;
elseif in == 1 && in2 == 0;
    p6 = 0; p12 = pFluid2;
elseif in ~= 0 && in2 ~= 0;
    p6 = 0; p12 = 0;
elseif in == 2 && in2 == 0;
    p6 = 0; p12 = pBranch*IntRate;
end

% Probability for diags
if upleft == 0 && upleft2 == 0;
    p13 = pFluidDiag; p25 =pFluidDiag2;
elseif upleft ==0 && upleft2 ~= 0;
    p13 = pFluidDiag; p25 = 0;
elseif upleft == 1 && upleft2 == 0;
    p13 = 0; p25 = pFluidDiag2;
elseif upleft ~= 0 && upleft2 ~= 0;
    p13 = 0; p25 = 0;
elseif upleft == 2 && upleft2 == 0;
    p13 = 0; p25 = pBranch*IntRate;
end

if upright == 0 && upright2 == 0;
    p14 = pFluidDiag; p26 = pFluidDiag2;
elseif upright ==0 && upright2 ~= 0;
    p14 = pFluidDiag; p26 = 0;
elseif upright == 1 && upright2 == 0;
    p14 = 0; p26 = pFluidDiag2;
elseif upright ~= 0 && upright2 ~= 0;
    p14 = 0; p26 = 0;
elseif upright == 2 && upright2 == 0;
    p14 = 0; p26 = pBranch*IntRate;
end

if upout == 0 && upout2 == 0;
```

```
    p15 = pFluidDiag; p27 = pFluidDiag2;
elseif upout ==0 && upout2 ~= 0;
    p15 = pFluidDiag; p27 = 0;
elseif upout == 1 && upout2 == 0;
    p15 = 0; p27 =pFluidDiag2;
elseif upout ~= 0 && upout2 ~= 0;
    p15 = 0; p27 = 0;
elseif upout == 2 && upout2 == 0;
    p15 = 0; p27 = pBranch*IntRate;
end

if upin == 0 && upin2 == 0;
    p16 = pFluidDiag; p28 = pFluidDiag2;
elseif upin ==0 && upin2 ~= 0;
    p16 = pFluidDiag; p28 = 0;
elseif upin == 1 && upin2 == 0;
    p16 = 0; p28 = pFluid2*pFluidDiag2;
elseif upin ~= 0 && upin2 ~= 0;
    p16 = 0; p28 = 0;
elseif upin == 2 && upin2 == 0;
    p16 = 0; p28 = pBranch*IntRate;
end

if leftout == 0 && leftout2 == 0;
    p17 = pFluidDiag; p29 = pFluidDiag2;
elseif leftout ==0 && leftout2 ~= 0;
    p17 = pFluidDiag; p29 = 0;
elseif leftout == 1 && leftout2 == 0;
    p17 = 0; p29 = pFluidDiag2;
elseif leftout ~= 0 && leftout2 ~= 0;
    p17 = 0; p29 = 0;
elseif leftout == 2 && leftout2 == 0;
    p17 = 0; p29 = pBranch*IntRate;
end

if leftin == 0 && leftin2 == 0;
    p18 = pFluidDiag; p30 = pFluidDiag2;
elseif leftin ==0 && leftin2 ~= 0;
    p18 =pFluidDiag; p30 = 0;
elseif leftin == 1 && leftin2 == 0;
    p18 = 0; p30 = pFluidDiag2;
elseif leftin ~= 0 && leftin2 ~= 0;
    p18 = 0; p30 = 0;
elseif leftin == 2 && leftin2 == 0;
    p18 = 0; p30 = pBranch*IntRate;
end

if rightout == 0 && rightout2 == 0;
    p19 = pFluidDiag; p31 = pFluidDiag2;
elseif rightout ==0 && rightout2 ~= 0;
    p19 = pFluidDiag; p31 = 0;
elseif rightout == 1 && rightout2 == 0;
    p19 = 0; p31 = pFluidDiag2;
elseif rightout ~= 0 && rightout2 ~= 0;
```

```
    p19 = 0; p31 = 0;
elseif rightout == 2 && rightout2 == 0;
    p19 = 0; p31 = pBranch*IntRate;
end

if rightin == 0 && rightin2 == 0;
    p20 = pFluidDiag; p32 = pFluidDiag2;
elseif rightin ==0 && rightin2 ~= 0;
    p20 = pFluidDiag; p32 = 0;
elseif rightin == 1 && rightin2 == 0;
    p20 = 0; p32 = pFluidDiag2;
elseif rightin ~= 0 && rightin2 ~= 0;
    p20 = 0; p32 = 0;
elseif rightin == 2 && rightin2 == 0;
    p20 = 0; p32 = pBranch*IntRate;
end

if downleft == 0 && downleft2 == 0;
    p21 = pFluidDiag; p33 = pFluidDiag2;
elseif downleft ==0 && downleft2 ~= 0;
    p21 = pFluid*pFluidDiag; p33 = 0;
elseif downleft == 1 && downleft2 == 0;
    p21 = 0; p33 = pFluidDiag2;
elseif downleft ~= 0 && downleft2 ~= 0;
    p21 = 0; p33 = 0;
elseif downleft == 2 && downleft2 == 0;
    p21 = 0; p33 = pBranch*IntRate;
end

if downright == 0 && downright2 == 0;
    p22 = pFluidDiag; p34 = pFluidDiag2;
elseif downright ==0 && downright2 ~= 0;
    p22 = pFluidDiag; p34 = 0;
elseif downright == 1 && downright2 == 0;
    p22 = 0; p34 = pFluidDiag2;
elseif downright ~= 0 && downright2 ~= 0;
    p22 = 0; p34 = 0;
elseif downright == 2 && downright2 == 0;
    p22 = 0; p34 = pBranch*IntRate;
end

if downout == 0 && downout2 == 0;
    p23 = pFluidDiag; p35 = pFluidDiag2;
elseif downout ==0 && downout2 ~= 0;
    p23 = pFluidDiag; p35 = 0;
elseif downout == 1 && downout2 == 0;
    p23 = 0; p35 = pFluidDiag2;
elseif downout ~= 0 && downout2 ~= 0;
    p23 = 0; p35 = 0;
elseif downout == 2 && downout2 == 0;
    p23 = 0; p35 = pBranch*IntRate;
end

if downin == 0 && downin2 == 0;
```

```
    p24 = pFluidDiag; p36 = pFluidDiag2;
elseif downin ==0 && downin2 ~= 0;
    p24 = pFluidDiag; p36 = 0;
elseif downin == 1 && downin2 == 0;
    p24 = 0; p36 = pFluidDiag2;
elseif downin ~= 0 && downin2 ~= 0;
    p24 = 0; p36 = 0;
elseif downin == 2 && downin2 == 0;
    p24 = 0; p36 = pBranch*IntRate;
end

ProbMat= abs(DirecMat-1); % Characterizes the probability of a hop to this site
    kIon =
[p1;p2;p3;p4;p5;p6;p7;p8;p9;p10;p11;p12;p13;p14;p15;p16;p17;p18;p19;p20;p21;p22;p23;p24;p25;p26;p
27;p28;p29;p30;p31;p32;p33;p34;p35;p36];
    HopProb = ProbMat.*kIon;
    r=rand(1);
    ProbSum=cumsum(HopProb);
    MoveSelect=r*ProbSum(6);
    MoveFinal=find(ProbSum >= MoveSelect,1,'first');

% Defining/storing diffusional positions per time step
Direc2=struct();
    % Nearest neighbors (6 in SC lattice)
    Direc2.up=VolMat2(Position.xi-1,Position.yi,Position.zi,step);
    Direc2.down=VolMat2(Position.xi+1,Position.yi,Position.zi,step);
    Direc2.right=VolMat2(Position.xi,Position.yi+1,Position.zi,step);
    Direc2.left=VolMat2(Position.xi,Position.yi-1,Position.zi,step);
    Direc2.out=VolMat2(Position.xi,Position.yi,Position.zi-1,step);
    Direc2.in=VolMat2(Position.xi,Position.yi,Position.zi+1,step);
    %next-nearest neighbors
    Direc2.up2=VolMat2(Position.xi-1,Position.yi,Position.zi,step);
    Direc2.down2=VolMat2(Position.xi+1,Position.yi,Position.zi,step);
    Direc2.right2=VolMat2(Position.xi,Position.yi+1,Position.zi,step);
    Direc2.left2=VolMat2(Position.xi,Position.yi-1,Position.zi,step);
    Direc2.out2=VolMat2(Position.xi,Position.yi,Position.zi-1,step);
    Direc2.in2=VolMat2(Position.xi,Position.yi,Position.zi+1,step);
    % First diagonal neighbors
    Direc2.upleft=VolMat2(Position.xi-1,Position.yi-1,Position.zi,step);
    Direc2.upright=VolMat2(Position.xi-1,Position.yi+1,Position.zi,step);
    Direc2.upout=VolMat2(Position.xi-1,Position.yi,Position.zi-1,step);
    Direc2.upin=VolMat2(Position.xi-1,Position.yi,Position.zi+1,step);
    Direc2.leftout=VolMat2(Position.xi,Position.yi-1,Position.zi-1,step);
    Direc2.leftin=VolMat2(Position.xi,Position.yi-1,Position.zi+1,step);
    Direc2.rightout=VolMat2(Position.xi,Position.yi+1,Position.zi-1,step);
    Direc2.rightin=VolMat2(Position.xi,Position.yi+1,Position.zi+1,step);
    Direc2.downleft=VolMat2(Position.xi+1,Position.yi-1,Position.zi,step);
    Direc2.downright=VolMat2(Position.xi+1,Position.yi+1,Position.zi,step);
    Direc2.downout=VolMat2(Position.xi+1,Position.yi,Position.zi-1,step)
    Direc2.downin=VolMat2(Position.xi+1,Position.yi,Position.zi+1,step);
    % Second diagonal neighbors
    Direc2.upleft2=VolMat2(Position.xi-2,Position.yi-2,Position.zi,step);
    Direc2.upright2=VolMat2(Position.xi-2,Position.yi+2,Position.zi,step);
    Direc2.upout2=VolMat2(Position.xi-2,Position.yi,Position.zi-2,step);
```

```
    Direc2.upin2=VolMat2(Position.xi-2,Position.yi,Position.zi+2,step);
    Direc2.leftout2=VolMat2(Position.xi,Position.yi-2,Position.zi-2,step);
    Direc2.leftin2=VolMat2(Position.xi,Position.yi-2,Position.zi+2,step);
    Direc2.rightout2=VolMat2(Position.xi,Position.yi+2,Position.zi-2,step);
    Direc2.rightin2=VolMat2(Position.xi,Position.yi+2,Position.zi+2,step);
    Direc2.downleft2=VolMat2(Position.xi+2,Position.yi-2,Position.zi,step);
    Direc2.downright2=VolMat2(Position.xi+2,Position.yi+2,Position.zi,step);
    Direc2.downout2=VolMat2(Position.xi+2,Position.yi,Position.zi-2,step);
    Direc2.downin2=VolMat2(Position.xi+2,Position.yi,Position.zi+2,step);
        if ProbSum~=0 % Tells the ion to move to a given position based on the chosen direction
            if MoveFinal== 1
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.up==1 ;
                    VolMat2(Position.xi-1,Position.yi,Position.zi,step)=1;
            elseif MoveFinal== 2
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.down==1;
                    VolMat2(Position.xi+1,Position.yi,Position.zi,step)=1;
            elseif MoveFinal== 3
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.right==1;
                    VolMat2(Position.xi,Position.yi+1,Position.zi,step)=1;
            elseif MoveFinal == 4
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.left==1;
                    VolMat2(Position.xi,Position.yi-1,Position.zi,step)=1;
            elseif MoveFinal == 5
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.out==1;
                    VolMat2(Position.xi,Position.yi,Position.zi-1,step)=1;
            elseif MoveFinal == 6
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.in==1;
                    VolMat2(Position.xi,Position.yi,Position.zi+1,step)=1;
            elseif MoveFinal == 7
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.up2==1;
                    VolMat2(Position.xi-2,Position.yi,Position.zi,step)=1;
            elseif MoveFinal == 8
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.down2==1;
                    VolMat2(Position.xi+2,Position.yi,Position.zi,step)=1;
            elseif MoveFinal ==9
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.right2==1;
                    VolMat2(Position.xi,Position.yi+2,Position.zi,step)=1;
            elseif MoveFinal == 10
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.left2==1;
                    VolMat2(Position.xi,Position.yi-2,Position.zi,step)=1;
            elseif MoveFinal == 11
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.out2==1;
                    VolMat2(Position.xi,Position.yi,Position.zi-2,step)=1;
            elseif MoveFinal == 12
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.in2==1;
                    VolMat2(Position.xi,Position.yi,Position.zi+2,step)=1;
            elseif MoveFinal == 13
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upleft==1;
                VolMat2(Position.xi-1,Position.yi-1,Position.zi);
            elseif MoveFinal == 14
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upright==1;
                VolMat2(Position.xi-1,Position.yi+1,Position.zi);
            elseif MoveFinal == 15
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upout==1;
```

```
            VolMat2(Position.xi-1,Position.yi,Position.zi-1);
        elseif MoveFinal == 16
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upin==1;
            VolMat2(Position.xi-1,Position.yi,Position.zi+1);
        elseif MoveFinal == 17
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.leftout==1;
            VolMat2(Position.xi,Position.yi-1,Position.zi-1);
        elseif MoveFinal == 18
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.leftin==1;
            VolMat2(Position.xi,Position.yi-1,Position.zi+1);
        elseif MoveFinal == 19
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.rightout==1;
            VolMat2(Position.xi,Position.yi+1,Position.zi-1);
        elseif MoveFinal == 20
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.rightin==1;
            VolMat2(Position.xi,Position.yi+1,Position.zi+1);
        elseif MoveFinal == 21
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downleft==1;
            VolMat2(Position.xi+1,Position.yi-1,Position.zi);
        elseif MoveFinal == 22
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downright==1;
            VolMat2(Position.xi+1,Position.yi+1,Position.zi);
        elseif MoveFinal == 23
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downout==1;
            VolMat2(Position.xi+1,Position.yi,Position.zi-1);
        elseif MoveFinal == 24
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downin==1;
            VolMat2(Position.xi+1,Position.yi,Position.zi+1);
        elseif MoveFinal == 25
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upleft2==1;
            VolMat2(Position.xi-2,Position.yi-2,Position.zi);
        elseif MoveFinal == 26
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upright2==1;
            VolMat2(Position.xi-2,Position.yi+2,Position.zi);
        elseif MoveFinal == 27
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upout2==1;
            VolMat2(Position.xi-2,Position.yi,Position.zi-2);
        elseif MoveFinal == 28
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.upin2==1;
            VolMat2(Position.xi-2,Position.yi,Position.zi+2);
        elseif MoveFinal == 29
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.leftout2==1;
            VolMat2(Position.xi,Position.yi-2,Position.zi-2);
        elseif MoveFinal == 30
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.leftin2==1;
            VolMat2(Position.xi,Position.yi-2,Position.zi+2);
        elseif MoveFinal == 31
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.rightout2==1;
            VolMat2(Position.xi,Position.yi+2,Position.zi-2);
        elseif MoveFinal == 32
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.rightin2==1;
            VolMat2(Position.xi,Position.yi+2,Position.zi+2);
        elseif MoveFinal == 33
            VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downleft2==1;
```

```
                VolMat2(Position.xi+2,Position.yi-2,Position.zi);
            elseif MoveFinal == 34
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downright2==1;
                VolMat2(Position.xi+2,Position.yi+2,Position.zi);
            elseif MoveFinal == 35
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downout2==1;
                VolMat2(Position.xi+2,Position.yi,Position.zi-2);
            elseif MoveFinal == 36
                VolMat2(Position.xi,Position.yi,Position.zi,step) = 0 && Direc2.downin2==1;
                VolMat2(Position.xi+2,Position.yi,Position.zi+2);
            end
                else
                tot=0; AA=0;
        end
```

```
    end
end
VolMat=VolMat2(:,:,:,step); % Redefines the initial matrix at the beginning of the for loop per
step to track ion motion
% To plot the ion distribution over time
 InDendrimer = find(VolMat2(:,:,(8:12),step)==1);    % Finds all ions in the z-range defined as
the dendrimer film
 NumberInDendrimer=length(InDendrimer)
 ThroughDendrimer=find(VolMat2(:,:,(1:7),step)==1);  % Finds all ions in the z-range defined as
below the dendrimer film
 NumberThroughDendrimer=length(ThroughDendrimer)

 hold on % Plots the ion distribution (in and through the film) as a function of time
 figure(4);
    if NumberInDendrimer ~=0;
    inD=find(VolMat2(:,:,(8:12),step)==1);
    NinD=length(inD);
    elseif NumberInDendrimer==0;
     NinD=0
    end
    if NumberThroughDendrimer ~= 0;
    thruD=find(VolMat2(:,:,(1:7),step)==1);
    NthruD=length(thruD);
    elseif NumberThroughDendrimer == 0;
        NthruD=0;
    end
 step
 plot(step,NinD,'-r','LineWidth',5);
 plot(step,NthruD,'-b','LineWidth',5);
 axis([0 stepmax 0 20]);
  legend('Ion Count Through','Ion Count In');
 xlabel('Time Step (fs)');
 ylabel('Number of Ions');
 title('Diffused Ion Distribution');
 grid on;
 hold off

%  if step > 1
%      VolMat2(:,:,:,step-1)=[];
```

```
%  end

 end
```

*Diffused ion counting and plotting*

Displays the final counts of ions that diffused into and through the dendrimer film

```
InDendrimer = find(VolMat2(:,:,(8:12),stepmax)==1);
NumberInDendrimer=length(InDendrimer)
ThroughDendrimer=find(VolMat2(:,:,(1:7),stepmax)==1);
NumberThroughDendrimer=length(ThroughDendrimer)
```

*Published with MATLAB® R2013a*

**References:**

"On the structure of polyaminoamide dendrimer monolayers," V. Bliznyuk et al., *Polymers* 39, 5249-5252 (1998)

Polyamidoamine (PAMAM) Dendrimer; Dendritech, Inc.: Midland, MI, June 8, 2011, http://www.dendritech.com/PAMAM%20MSDS%20WATER.pdf (accessed May 6, 2013)

Methanol; MSDS No. 322415 [Online]; Sigma-Aldrich: Saint Louis, MO, February 8, 2013, http://www.sigmaaldrich.com/MSDS/MSDS/DisplayMSDSPage.do?country=US&language=en&productNumber=322415&brand=SIAL&PageToGoToURL=http%3A%2F%2Fwww.sigmaaldrich.com%2Fcatalog%2Fproduct%2Fsial%2F322415%3Flang%3Den, (accessed May 6, 2013)

"Ringer's solution recipe," A. M. Helmenstine, *About.com, Chemistry* (2013)

"Diffusion of ions in seawater and in deep-sea sediment," Y.-H. Li and S. Gregory, *Geochemica et Cosmochemica Acta* 38, 708-713 (1974)

"An overview of the corrosion aspect of dental implants (titanium and its alloys)," T. P. Chaturvedi, *Indian Journal of Dental Research* 20, 91-98 (2009)

"External Electrostatic Interaction versus Internal Encapsulation between Cationic Dendrimers and Negatively Charged Drugs: Which Contributes More to Solubility Enhancement of the Drugs?" Y. Cheng et al., *The Journal of Physical Chemistry B.* 112, 8884-8890 (2008)

"Dendrimer Encapsulated Metal Nanoparticles: Synthesis, Characterization, and Applications to Catalysis," R. Crooks, *Accounts of Chemical Research* 34, 181-190 (2001)

"Decreased bacterial adhesion to surface-treated titanium," B. Del Curto et al., *International journal of artificial organs* 28, 718-730 (2005)

"Dendrimers: a review of their appeal and applications," G. M. Dykes, *J. Chem. Technol. Biotechnol.* 76, 903–918 (2001)

"Poly(amidoamine) (PAMAM) dendrimers: from biomimicry to drug delivery and biomedical applications," R. Esfand et al., *Drug Discovery Today* 6, 427-436 (2001)

"Interaction forces between poly(amidoamine) (PAMAM) dendrimers adsorbed on gold surfaces," D. Hiraiwa, *Journal of Colloid Interface Science* 298, 982-986 (2006)

"Functional Dendrimers, Hyperbranched and Star Polymers," K. Inoue, *Progress in Polymer Science*, 453-571 (2000)

"Electron Capture Dissociation and Collision-Induced Dissociation of Metal Ion (Ag+, Cu2+, Zn2+, Fe2+, and Fe3+) Complexes of Polyamidoamine (PAMAM) Dendrimers," M. Kaczorowska, *Journal of the American Society of Mass Spectrometry* 4, 674-681 (2009)

"Surface Spectroscopic Characterization of Titanium Implant Materials," J. Lausmaa, B. Kasemo, and H. Mattsson, *Applied Surface Science* 44, 133-146 (1990)

"Efficiency of various lattices from hard ball to soft ball: theoretical study of thermodynamic properties of dendrimer liquid crystal from atomistic simulation," Y. Li et al., *J Am Chem Soc.* 126, 1872-1885 (2004)

"Biomedical Implants: Corrosion and its Prevention - A Review," G. Manivasagam, *Recent Patents on Corrosion Science* 2, 40-54 (2010)

"Surface adsorption of model dendrimers," M. Mansfield et al.,. *Polymer* 37, 3835-3841 (1995)

"Dendrimers—Branching Out from Curiosities into New Technologies," O. Matthews et al., *Prog Polym Sci* 23, 1-56 (1998)

"Mechanical biocompatibilities of titanium alloys for biomedical applications," M. Niinomi, *Journal of the Mechanical Behavior of Biomedical Materials* 1, 30-42 (2008)

"Corrosion resistance for biomaterial applications of $TiO_2$ films deposited on titanium and stainless steel by ion-beam-assisted sputtering," J. Pan et al., *Journal of Biomedical Materials Research* 35, 309-318 (1998)

"Counterion Distribution and Zeta-Potential in PAMAM Dendrimer," K. P. Maiti et al., *Macromolecules* 41, 5002-5006 (2008)

"Electrochemical impedance spectroscopy: an overview of bioanalytical applications," E. Randviir et al., *Anal. Methods* 5, 1098 (2013)

"Metallic Medical Implants: Electrochemical Characterization of Corrosion Processes," P. Schmutz, N. C. Quach-Vu, and I. Gerber, *Electrochemical Society Interface*, 35-40 (2008)

"Corrosion degradation and prevention by surface modification of biometallic materials," R. Singh and N. Dahotre, *J. Mater. Sci: Mater Med.* 18, 725-751 (2007)

"Stability, Antimicrobial Activity, and Cytotoxicity of Poly(amidoamine) Dendrimers on Titanium Substrates," L. Wang, *Applied Materials and Interfaces* 3, 2885-2894 (2011)

"Compositional and structural evolution of the titanium dioxide formation by thermal oxidation," S. Wei-Feng et al., *Chinese Physics B* 17, 3003-3005 (2008)

"Surface studies of in vitro biocompatibility of titanium oxide coatings," M. P. Casaletto et al., *Applied Surface Science* 172, 167-177 (2001)

"The Effect of Surface Roughness on Early In Vivo Plaque Colonization on Titanium," L. Rimondini et al., *J. Periodontol* 68, 556-562 (1997)

"Low-Temperature Titanium-Based Wafer Bonding," J. Yu et al., *Journal of The Electrochemical Society* 154, H20-H25 (2007)

"Unalloyed titanium for implants in bone surgery," O. Pohler, *Injury* 31, D7-D13 (2000)

"In vivo metallic biomaterials and surface modification," T. Hanawa, *Materials Science and Engineering A* 267, 260-266 (1999)

"Local accumulation of titanium released from a titanium implant in the absence of wear," P. D. Bianco, P. Ducheyne, and J. M. Cuckler, *Journal of Biomedical Materials Research* 31, 227-234 (1996)

"HPLC analysis of PAMAM dendrimer based multifunctional devices," M. Islam, I. Majoros, and J. Baker, Jr., *Journal of Chromatography B* 822, 21-16 (2005)

"The structure and thermal stability of $TiO_2$ grown by the plasma oxidation of sputtered metallic Ti films," G. He et al., *Chemical Physics Letters* 395, 259-263 (2004)

"Electrochemical characterisation of biomedical alloys for surgical implants in simulated body fluids," C. Vidal and A. Muñoz, *Corrosion Science* 50, 1954-1961 (2008)

"Effect of deposition parameters and post-deposition annealing on the morphology and cellular response of electrosprayed $TiO_2$ films," T. Sebbowa et al., *Biofabrication* 3, 045001 (2011)

"Surface Aspects of Titanium Dental Implants," K. Turzo, *Biotechnology: Molecular Studies and Novel Applications for Improved Quality of Human Life,* ch. 9, 135-158 (2012)

"Adhesion, Growth and Differentiation of Osteoblasts on Surface-Modified Materials Developed for Bone Implants," M. Vandrovcová and L. Bačáková, *Physiol. Res.* 60, 403-417 (2011)

"The anatase phase of nanotopography titania plays an important role on osteoblast cell morphology and proliferation," J. He et al., *J. Mater. Sci.: Mater. Med.* 19, 3465-3472 (2008)

"Characteristics of the surface oxides on turned and electrochemically oxidized pure titanium implants up to dielectric breakdown: the oxide thickness, micropore configurations, surface roughness, crystal structure and chemical composition," Y. Sul et al., *Biomaterials* 23, 491-501 (2002)

"Hemocompatibility of titanium oxide films," N. Huang et al., *Biomaterials* 24, 2177-2187 (2003)

"Pitting Corrosion of Titanium: The Relationship Between Pitting Potential and Competitive Anion Adsorption at the Oxide Film/Electrolyte Interface," S. Basame and H. White, *Journal of The Electrochemical Society* 147, 1376-1381 (2000)

"Origins of Pitting Corrosion," G. T. Burstein et al., *Corrosion Engineering, Science and Technology* 39, 25-30 (2004)

"Structural evolution and optical properties of $TiO_2$ thin films prepared by thermal oxidation of sputtered Ti films," C. Ting and S. Chen, *Journal of Applied Physics* 88, 4628-4633 (2000)

"Spectroscopic ellipsometry on thin titanium oxide layers grown on titanium by plasma oxidation," G. Droulers et al., *Journal of Vacuum Science and Technology B* 29, 1071-1023 (2011)

"Dendrimer biocompatibility and toxicity," R. Duncan and L. Izzo, *Advanced Drug Delivery Reviews* 57, 2215-2237 (2005)

"Adsorption, Diffusion and Desorption of Chlorine on and from Rutile $TiO_2$ {110}: A Theoretical Investigation." O. Inderwildi et al., *ChemPhysChem.* 8 (3). (2007).

"Comparative Evaluation of Metal Ions Release from Titanium and $Ti_6Al_7Nb$ into Bio-Fluids," L. Josephs, O. Israel, and E. Edet, *Dental Research Journal* 6, 7-11 (2009).

"Galvanic corrosion behavior of titanium implants coupled to dental alloys," M. Cortada et al., *Journal of Materials Science: Materials in Medicine* 11, 287-293 (2000)