Modeling Synapse Formation and Growth as a Non-Biological Analog to the Brain

by

Sara V. Fernandez

Submitted to the Department of Materials Science and Engineering
in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science at the
Massachusetts Institute of Technology – June 2023

© 2023 Sara V. Fernandez
All rights reserved

The author hereby grants to MIT permission to reproduce and to
distribute publicly paper and electronic copies of this thesis document in whole or in part in any
medium now known or hereafter created.

Signature of Author .....

Sara V. Fernandez
Department of Materials Science and Engineering
May 23, 2023

Certified by .....

W. Craig Carter
Thesis Supervisor
Toyota Professor of Materials Processing

Accepted by .....

James LeBeau
Associate Professor of Materials Science and Engineering
Chair, DMSE Undergraduate Committee

Modeling synapse formation and growth as a non-biological analog to the brain

by

Sara V. Fernandez

Submitted to the Department of Materials Science and Engineering
in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science at the
Massachusetts Institute of Technology – June 2023

ABSTRACT

Efficiency in computing systems is a pressing concern as global reliance on machines and automation grows. Leveraging an understanding of the brain's exceptional computational capabilities, this study presents a preliminary nondimensionalized model of synapses, an essential component for developing brain-inspired computing systems. The model simulates a physical analog of synapse formation wherein a single two-nanowire junction in an electrolytic medium undergoes an electric potential, causing electric field-driven ion transport and subsequent filament growth. Simulations allow for the extraction of meaningful parameter relationships as well as governing equations relating both filament length and time, and current and time. By investigating electric potential-driven cation diffusion, the model provides insights for designing more advanced computing technologies. Future directions involve refining assumptions, adapting system geometry for dendritic growth, and modeling an entire nanowire network. This research bridges the gap between brain-inspired and physical computing, paving the way for highly efficient computing systems beyond traditional approaches.

Thesis supervisor: W. Craig Carter

Title: Toyota Professor of Materials Processing

ACKNOWLEDGEMENTS

I would like to thank Dr. W. Craig Carter for his mentorship, guidance with computational methods, and for assisting with troubleshooting during various points in the development of this work. Our discussions regarding fundamental materials science concepts as well as the value of nondimensionalization have greatly influenced this model's versatility.

I would additionally like to thank Dr. Jin-Hoon Kim for providing guidance throughout the duration of this project. I would specifically like to thank him for engaging in fruitful discussions on the driving forces of filament formation, assisting with mobility calculations, and capturing HR-SEM images of the physical nanowire system he developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SUPPLEMENTAL INFORMATION

Supplementary Video 1: Nondimensionalized filament growth for radius = 0.025
Supplementary Video 2: Nondimensionalized filament growth for radius = 0.05
Supplementary Video 3: Nondimensionalized filament growth for radius = 0.075

1. INTRODUCTION

The brain exhibits unmatched computational efficiency and has drawn widespread, multidisciplinary interest in recent years. The brain is made up of 86 billion neurons[1] which send information through a neural superhighway in the form of action potentials. The vast functionality of the brain positions it to perform complex operations on incoming sensory inputs, store and retrieve memories, and generate output in the form of thoughts, behaviors, and actions. The underlying mechanisms that dictate the brain's efficiency are largely structural[2–4]; however most modern attempts to replicate brain-like computing rely on algorithmic approaches which fail to harness the built-in connectivity of the brain's physical network.[5] While machine learning-based algorithms can significantly increase computing efficiency, computers still lag behind the immense efficiency of the brain. Therefore, it is vital to develop an approach for systematically building physical solutions to approach the human brain's $(5.52 \pm 1.13) \cdot 10^{16}$ bits/s and bridge the efficiency gap.[6]

The application of a voltage to a two-nanowire system within a dielectric medium has been demonstrated to exhibit similar self-assembly characteristics to those that make the brain so efficient.[7] With the application of voltage, a filament with the same composition as the positively charged nanowire will grow in accordance with the electric field, closing the gap between the two nanowires over time. Thus, this work will introduce a preliminary computational model that can inform the design of such

physical systems by predicting the filament's growth behavior in relation to various adjustable parameters.

2. BACKGROUND

2.1 The Human Brain

The brain is a highly efficient computational machine. It is composed of interconnected neurons that form connective pathways through which electrical signals can be sent. These signals carry information throughout the brain that influence how the body operates. Once a neuron is formed, it migrates to its designated position and extends axons in the direction of adjacent neurons. These axons segment into dendrites, or branching structures, that extend towards nearby neurons.[8] This neuron growth and subsequent formation of synapses, or synaptogenesis, is driven by both chemical and electrical signaling factors.[9] Once these neurons make contact with each other, a physical connection is formed that allows for a rapid transmission of electrical impulses through the junction between two neurons.

Synapses and other connective mechanisms allow the brain to achieve an efficiency that is six orders of magnitude greater than the most powerful computer in the world.[6] In order to harness these mechanisms, algorithmic approaches have been used with the aim of applying the brain's efficiency to modern day computing. Although it is imperative to span the efficiency gap, neural nets overlook several inherent

efficiencies of physical systems. The 3D geometry of neurons, for instance, allows them to form points of connection that are largely unreachable on a 2D transistor.[10]

2.2 A Physical Analog of Neuron Growth

Thus, it is imperative to develop a physical analog to the brain that can be used in applications of computing and beyond. To do this, a model that can harness the self-assembly capabilities of brain-like materials must first be created. A single two-nanowire junction can be modeled physically and computationally during the course of an applied voltage as an analog for synapse formation.[7,11] A visual comparison between a single two-nanowire junction (**Figure 1a**) and the corresponding simulated model constructed in this work (**Figure 1b**) are shown in **Figure 1**. In the brain, adjacent neurons experience an electrochemical driving force towards each other, causing axons and dendrites to spread out accordingly. This axon-and-dendrite growth can be represented as a two-nanowire junction that experiences an applied voltage, causing a filament to form through a dielectric medium between the two.[12]

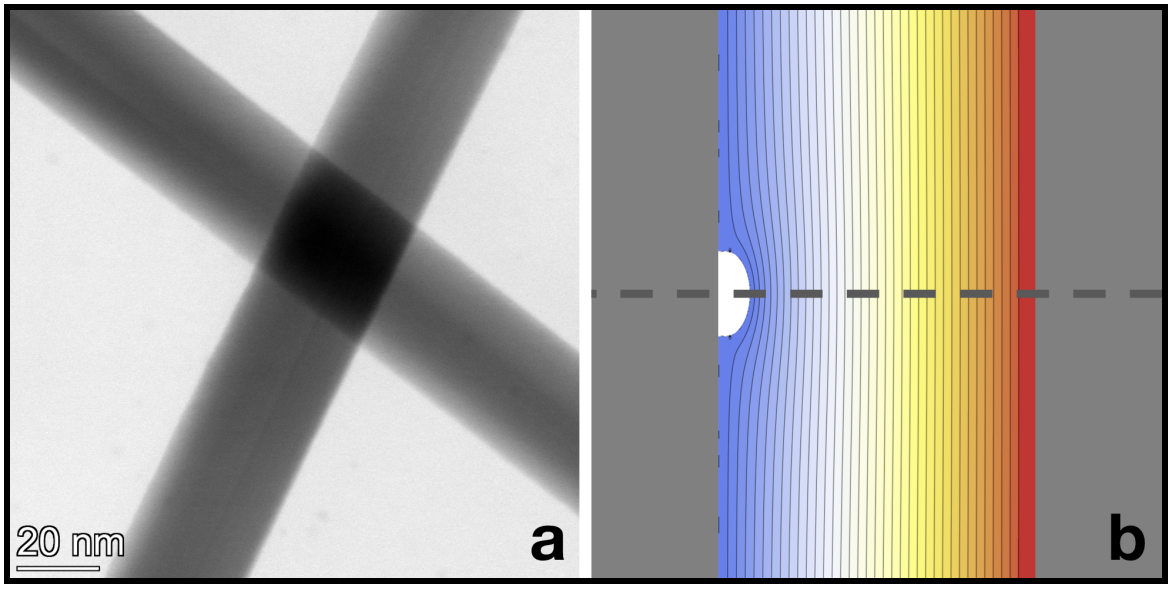


Figure 1: HR-TEM image and computational representation of a single two-nanowire junction. (**a**) High-resolution transmission electron microscopy (HR-TEM) image of a single two-nanowire junction. These two Ag nanowires on the Cu grid were prepared as described in Figure 8 as part of a nanowire network. (**b**) Analogous computational model of the 2D nanowire junction developed in this work. The gray rectangles on the left and right sides of the image represent the two nanowires. The white, curved shape on the leftmost nanowire represents the growing filament. The color gradient represents the electric potential through the dielectric medium. The horizontal dashed line is an axis of symmetry. All subsequent visualizations will depict only the top half of the filament and the dielectric medium it grows through for simplicity. Scale bar: 20 nm.

2.3 Fundamental Materials Properties and Behavior of Nanowire Systems

Silver (Ag) nanowires, for instance, demonstrate this filament formation in response to an applied voltage as well as exhibit volatile resistive switching under low electric fields.[13,14] Volatile resistive switching is the process by which, after the filament physically bridges the two nanowires and the circuit is shorted, a physical pathway for electrical current to pass through is retained in the form of the filament for a short duration. This self-assembly closely mimics self-assembly of neurons during synaptogenesis that corresponds with the formation of short-term memory in the brain. This volatile resistive memory allows the resistance of the material to be altered to represent binary data (0s and 1s) for use in computing. However, unlike traditional

resistive-switching random access memory (RRAM),[15] volatile resistive memory is designed to be volatile, meaning it loses its stored data soon after the removal of voltage, or once the power is turned off.[16] This differs from materials that exhibit non-volatile memory which retain the resistive memory for much longer durations in response to higher applied voltages. However, further consideration of non-volatility goes beyond the scope of this work.

Several nanowire systems in electrolytic mediums have been demonstrated to embody resistive memory and exhibit self-assembly behavior. With the application of a voltage to a nanowire, an electric potential is introduced such that one nanowire serves as a positive terminal and the other as a negative terminal. Due to this electric potential difference, metallic ions are oxidized from the nanowire with the greatest electric potential and reduced near the nanowire with the lowest electric potential.[17]

Once the cations enter the electrolyte, or dielectric medium, that separates the two nanowires, they undergo cation transport from the positive nanowire to the negative nanowire. This process occurs across the entire length of the nanowires, and in the case of two equivalent nanowires, the factors dictating ion transport remain consistent between both cations and anions. Although there are many different factors that influence this transport and subsequent filament growth, the electric field has been shown to be the dominant driver of ionic current.[17,18] The electric field will therefore be treated as the dominant driving force in the computational models and simulations to follow. This allows further simplification as the effects of mechanisms such as electron

charge transfer, ion diffusion, and ion hopping, among others, can be ignored. Consideration of critical nucleus formation can also be bypassed since the computational simulations in this work assume a nonzero initial filament length. Additionally, if the applied voltage is within ~ 0.7 V and 1.2 V, filament growth rate should not be nucleation-limited.[17] However, these aforementioned transport mechanisms are still present, so it remains valid to reference them for the calculation of certain materials parameters associated with individual system components.

Once cations traverse the gap separating the two nanowires, a filament begins to form as the cations are reduced and connect to the nanowire that experiences the lowest electric potential. As this process continues, the cations reduce and form a filament that grows towards the nanowire that experiences the greatest electric potential. When the filament reaches the nanowire, it causes a large increase in the current flowing in the system.

3. METHODS

3.1 Underlying Simplifications and Assumptions for the Simulation

Once the filament bridges the gap between the two nanowires, current can flow more freely from nanowire to nanowire. More precisely, once the filament tip becomes sufficiently close to the opposite nanowire, tunneling current can become the dominant mechanism for charge transport.[17] However, for the purposes of the preliminary simulation, the focus will be limited to the formation of the filament as well as the large

increase in conductivity that results. Additionally, the memristive behavior that follows the short-circuiting of the system and the subsequent removal of applied current will be left for future explorations.

To predict the behavior of this physical brain-emulating system, a preliminary computational simulation has been developed in Mathematica, Version 13.2.1.0. This models the two-dimensional growth of a metallic filament connecting a single two-nanowire junction that is subjected to an applied voltage with the initial state shown in **Figure 1a**. For simplicity, only one growing filament is modeled, and its geometry is assumed to be self-similar such that any growth corresponds to a translation of the leading end of the filament towards the nanowire ahead of it. The width is similarly held constant such that changes in filament area exclusively cause an increase in one-dimensional growth forward, towards the nanowire ahead of the growing filament. There is assumed to be a plane of symmetry through the tip, or leading point, of the growing filament, so the system is modeled using a half-width, or radius, in all instances following **Figure 1a**. The system is assumed to be held at room temperature at a pressure of 1 atm, and it is assumed to undergo no arbitrary physical perturbations from external influences.

The nanowires are also assumed to be conductive, infinitely long, and overlapping perpendicularly to each other. Additionally, the dielectric medium surrounding the nanowires is assumed to be homogeneous and isotropic. The electric field is also assumed to be electrostatic (uninfluenced by external magnetic fields) and

is considered to be the only electric field acting on the system. Although these idealized assumptions greatly simplify the system's mathematical modeling and reduce the computational load, it is important to note that removing some of these simplifying assumptions could allow for a more accurate, complex model.

For this simulation, the ion mobility μ_0 was treated as a constant since the materials within the system are homogenous and the electric field strength is low such that the analogy to volatile memory is preserved (i.e. with low applied voltage). When the velocity of charge carriers does not significantly depend on the strength of the electric field, such as with a low electric field strength, the mobility can be reasonably approximated as a constant. For a more sophisticated model, the dependence of mobility on dynamic factors such as carrier concentration and electric field strength should be considered. However, this is beyond the scope of this current simulation.

3.2 Building the Model's Initial State

3.2.1 Filament Geometry

Creating the model begins with selecting the geometry of the filament. The model visualizes a half-filament about the axis of symmetry shown in **Figure 1a**. The filament is modeled as having straight sides, a constant width equivalent to the radius, and a curved leading edge. The filament grows from one of the two nanowires which are separated by a total distance, d. The rectangular boundary of the filament is composed of a series of connected points with maximum y-values equivalent to the

radius. The leading, curved region of the filament is composed of a quarter circle created based on the input radius. Since the radius of the filament is equivalent to the width and is used to calculate the quarter-circular tip geometry, the initial filament length must be greater than the radius. These values can be changed as desired.

For the majority of the following simulations, the radius is selected to be equivalent to 5% of d, or equivalently, 5% of the length of the overlapping portions at each nanowire junction. This is denoted as a nondimensionalized length of 0.05. Similarly, the initial filament length is selected to be 10% of d and is denoted as having a nondimensionalized length of 0.1. A more detailed explanation of nondimensionalization is provided in the following section. More detailed geometric equations and visualized explanations can be found in the Mathematica Notebook excerpts in the **Appendix**.

3.2.2 Parameter Nondimensionalization

In order to speed up computation and make this model applicable to a wide range of physical systems, all variables are nondimensionalized. In order to put variables into a nondimensionalized form, the units must cancel out such that broad, factor-based relationships between variables can be uncovered. As a general rule-of-thumb, the nondimensionalized form of a variable can be reached by taking the ratio of the actual variable and the reference value, or initial condition.

In this model, the nondimensionalized form of several variables are presented, including position (x^{non} and y^{non}), electric potential (Φ^{non}), ion mobility (μ^{non}), time (τ^{non}), and current (I^{non}) as follows in **Equations 1-6**.

$$x^{non} = \frac{x}{d} \tag{1}$$

$$y^{non} = \frac{y}{d} \tag{2}$$

$$\Phi^{non} = \frac{\Phi}{V_{app}} \tag{3}$$

$$\mu^{non} = \tau \mu_0 N_A dV_{app} \tag{4}$$

$$\tau^{non} = \tau \mu_0 N_A dV_{app} \tag{5}$$

$$I^{non} = \frac{I}{\mu_0 N_A dV_{app}} \tag{6}$$

Avogadro's number[19] (N_A) is included to account for the molar units of electric flux. The distance between the two nanowires is represented here by d, and the applied electric potential in the form of voltage is represented here by V_{app}. For completeness, x, y, and d have units of meters (m), Φ and V_{app} have units of volts (V), τ has units of seconds (s), μ_0 has units of $m^2/(V \cdot s)$, and I has units of amperes (A).

3.2.3 Mesh Construction

Once the system geometry is established, the filament points are used to create a boundary mesh surrounding the system, followed by an area mesh as a foundation for the simulation. Regions that directly surround the filament as well as those in the path of the growing filament are meshed much more finely since these regions undergo rapid changes in electric field (i.e. at the tip) and require more precise calculation to account for this. It should be noted that there exists a minor issue with using Mathematica's interpolation function on meshes, resulting in unexpected behavior at certain positions. However, this issue is currently being patched in Mathematica and has no measurable effect on this model. An example of this mesh can be found in the Mathematica Notebook excerpts in the **Appendix**.

3.2.4 Electric Potential

Following the creation of the mesh, a voltage, or electric potential, is introduced to the system. This is done using Laplace's equation[20] which states that, within a source-free region, the Laplacian of the electric potential field is zero, meaning that the charges of anions and cations cancel. In order for the Laplacian equation to be considered valid for this case, the system is assumed to be in a steady-state condition, meaning that the electric field and the geometry of the system do not change with time. Since this model recalculates the electric potential at finite time values rather than sweeping continuously over time, these conditions are upheld.

Unique solutions to Laplace's equation can be determined if either the function's value is specified on all boundaries (Dirichlet boundary condition) or the normal derivative of the function is specified on all boundaries (Neumann boundary condition).[21,22] As such, both Newmann and Dirichlet boundary conditions were incorporated.

3.2.5 Electric Field and Flux

Electric field could then be calculated in both x and y directions from the second-order partial differential equation for electric potential. It could then be related to electric flux by the ion mobility in the following equation (**Equation 7**),

$$J = \mu_0 \nabla \Phi \tag{7}$$

where J is the flux in mol/(m^2·s), μ_0 is the ion mobility, and $\nabla\Phi$ is the gradient of the electric potential, or the electric field in V/m.[23] The negative sign that is typically included in this equation is neglected in order to simplify computation since the one-dimensional direction of filament growth is known.

Although the total flux is typically understood to be a sum of the flux based on diffusion (as in Fick's First Law[23]) and the flux based on the drift term, the electric field is assumed to be the limiting factor of filament growth, with each cation moving in the direction of a local electric field (i.e. high electric potential) proportionally by a factor of μ_0. As a result, the effects of diffusion can be disregarded without fundamentally altering the accuracy of the simulation.

3.3 Creating a Dynamic, Iterative Simulation

3.3.1 Time Incrementation

Once the initial state is achieved in the simulation, it is important to investigate the filament growth behavior as a function of time. To do so, a nondimensionalized time step Δt must be introduced. For the simulations and equation fitting presented in this work, a nondimensionalized time step of $\Delta t = 0.0001$ is used. (However, note that exported videos in the **Supplemental Information** use a nondimensionalized time step of $\Delta t = 0.001$ to reduce file size and increase exporting speed.)

3.3.2 Dynamic Calculation of Flux

With each time increment, there will be a change in flux as well as a subsequent change in filament length. The changes in flux over time can be calculated by integrating the flux within a circle that encompasses the entire filament and its surrounding electrolyte at discrete times separated by a small time step of Δt. In amorphous materials like electrolytes, the motion of ions is often described as being driven by ion hopping, but assuming the concentration of ions and temperature are held constant in this system, immediate reuptake becomes a reasonable assumption.[24] This means that the ion drift is the limiting factor wherein the electric field drives the ion transfer and that the flux directly impacts the quantity of reduced cations that are added to the tip of the growing filament.

3.3.3 Dynamic Calculation of Filament Length

Following this flux determination, the filament length is updated with a length value corresponding to the area increase associated with the change in flux. The magnitude of the flux is equivalent to the ionic current, or the count associated with the influx of cations ("area in"). The "area in" can be treated as equivalent to the flux times the nondimensionalized area of the atom of interest. Note that this disregards the effects of lattice packing and density on filament area. The nondimensionalized area of the atom of interest can be approximated to equal the flux since all atomic radii, r_{atom}, are several orders of magnitude smaller than the nondimensionalized distance, d, that separates the two nanowires. This approximation is used for this model and is shown in **Equation 8** below.

$$1 \sim \pi\left(\frac{r_{atom}}{d}\right)^2 \tag{8}$$

Thus, with each time step Δt, the flux is recalculated, causing the filament length to increase as the filament grows until the distance between nanowires, d, is reached or surpassed.

3.3.4 Tabulating Results for Analysis

Using this iterative approach, several values can be generated and tabulated, including time elapsed, filament length, filament area, and step area (or flux). An example creating these tables can be found in the Mathematica Notebook excerpts in

the **Appendix**. Note that tabulated values in the **Appendix** may show minute deviations from the results reported here due to re-running of the simulation. These values form the basis for the dynamic visualizations that model filament growth and allow for nonlinear data fitting such that quantitative relationships between both filament length and time, and current and time, can be extracted.

4. RESULTS AND DISCUSSION

4.1 Filament Length as a Function of Time

4.1.1 Nondimensionalization Approach

In order to assess the filament growth over time, a dynamic visualization was created wherein black flux vectors are generated and overlaid onto a color gradient that depicts the electric potential over time. This visualization utilized the tabulated values of time elapsed and filament length generated in previous steps. Snapshots depicting the filament growth over time are shown in **Figure 2a-c** with $\tau^{non} = 0$, $\tau^{non} = \tau^{non,\, tot/2}$, and $\tau^{non} = \tau^{non,tot}$, respectively. The corresponding video, as well as videos associated with different initial filament dimensions, can be found in the **Supplemental Information**. This simulation provides a visual representation of the underlying variable dependencies in this system.

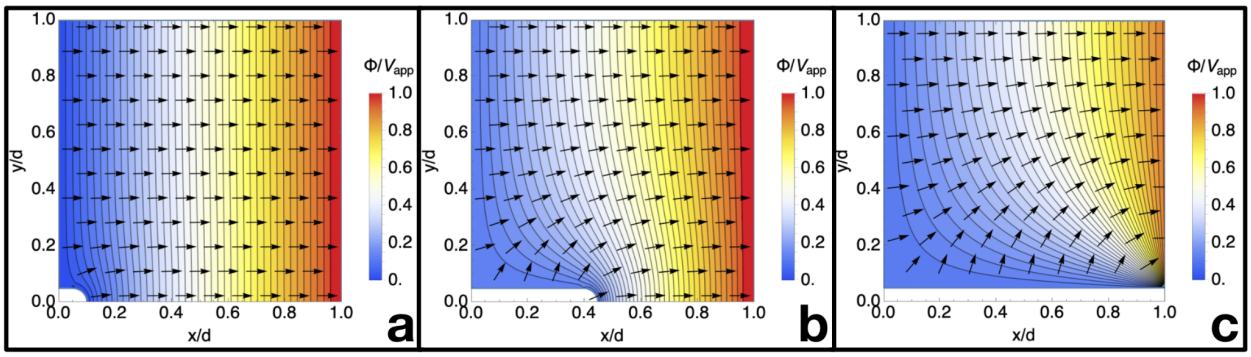


Figure 2: Simulation snapshots of nondimensionalized filament growth versus time. The color gradient represents the electric potential which increases from left to right, and the black arrows represent the electric field and electric flux. (**a**) Filament with initial length of 0.1 and initial radius of 0.05 at time $\tau^{non} = 0$. (**b**) Filament at time $\tau^{non} = \tau^{non,tot/2}$, once half of the time has passed. c. Filament at time $\tau^{non} = \tau^{non,tot}$, the first instance at which the filament has reached the rightmost nanowire. Note the changes in both the electric potential and arrow directions with the passage of time.

The filament begins with an initial length of 0.1 and radius of 0.05 in nondimensionalized form. This initial state can be seen in **Figure 2a**. With each nondimensionalized time step Δt (Δt = 0.0001 in this case), the filament grows towards the rightmost nanowire corresponding to the region of greatest electric potential, indicated in red. The image in **Figure 2b** captures the state at which half of the total time has elapsed. Note that the tip of the filament has not yet reached the halfway point (x/d = 0.55) between its initial position (x/d = 0.1) and the position of the rightmost nanowire (x/d = 1) despite the passage of half of the total time. This indicates that the growth rate increases as the filament approaches the region of greatest potential. Because of this, initial growth from $\tau^{non} = 0$ at **Figure 2a** to $\tau^{non} = \tau^{non,tot/2}$ at **Figure 2b** is shown to be smaller than between **Figure 2b** and **Figure 2c** at $\tau^{non} = \tau^{non,tot}$.

Additionally, the black vectors represent the magnitude of the flux present in the system. However, the arrows are flipped to point in the direction of the filament growth

resulting from the flux-driven addition of area. Since the electric field and flux are treated as equivalent with μ_0 assumed to be constant at 1, these vectors also depict the electric field profile. **Equation 7** shows the relationship between electric flux and electric field used here which neglects the negative sign that is typically associated with this relationship in favor of illustrating the direction toward which the growth is driven. The directions of these vectors change over time with greater flux concentrated at the leading end of the growing filament. The magnitudes of these vectors' angular changes directly describe the flux at any given time and collectively contribute to the area added onto the filament as it grows, supporting the simplification that the majority of the area will be added to the filament tip.

4.1.2 Determination of Governing Equation for Time-Dependent Filament Growth

These relationships can be quantified using nonlinear data fitting methods in Mathematica (see **Appendix**). The relationship between filament length and time for this case (with initial filament length of 0.1 and width of 0.05) has been found to be described by **Equation 9**.

$$x = \left(0.0593429 exp\left(15.3971 \sqrt{\tau \mu_0 N_A dV_{app}} \right) + 0.0286777 \right) d \tag{9}$$

Constants in this nonlinear exponential equation were found to be a = 0.0593429, b = 15.3971, and c = 0.028677. **Equation 9** has an adjusted R^2 value of 0.999814 and parameter confidence intervals[25] with a confidence level of 95% for a, b, and c, respectively, as follows: (0.0571059, 0.0615798), (15.2039, 15.5902), and

(0.0237774, 0.033578). This means that the true parameters for a, b, and c are 95% likely to lie within the specified ranges. **Equation 9** is plotted in **Figure 3** along with the relevant data. Because the nondimensionalized time step Δt = 0.0001 is so small, it may be difficult to distinguish between the different data points. The thick, half-opacity blue curve consists of these data points, and the thin, full-opacity blue curve that overlaps with it is **Equation 9**.

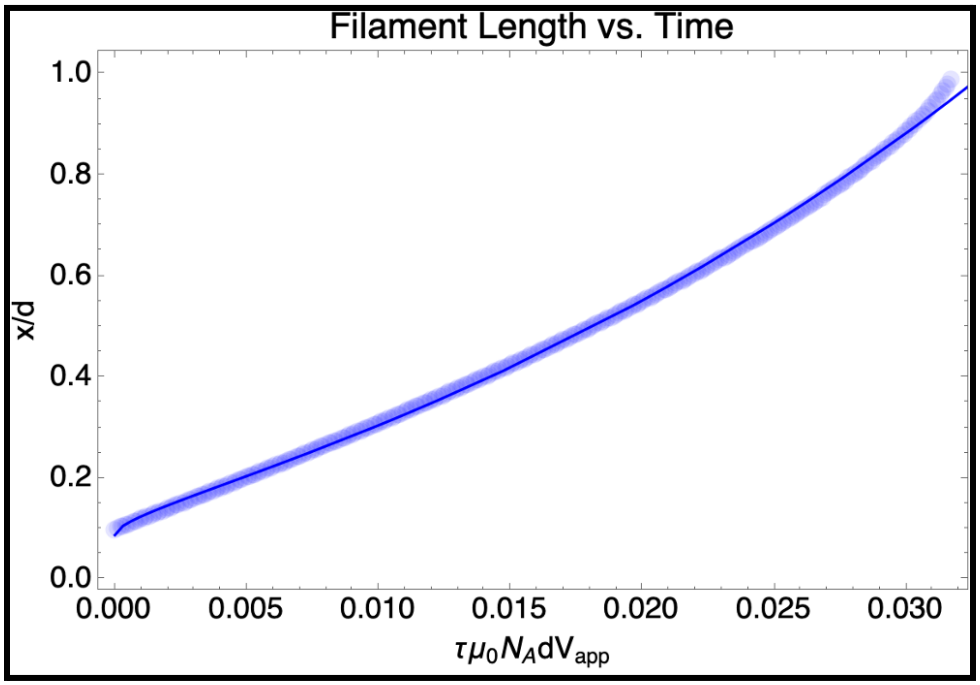


Figure 3: Fitted curve of nondimensionalized filament length versus time. The filament has an initial length of 0.1 and a radius of 0.05. The thick, half-opacity blue curve is composed of computationally-generated data points. The thin, full-opacity blue curve fits this data and is described by **Equation 9**. Note the nonlinear exponential relationship.

Using the value for τ^{non} in **Figure 3** at which $\tau^{non} = \tau^{non,tot}$ and the x-axes conversion between τ^{non} and τ from **Equation 5**, several parameter relationships become clear. The applied voltage (V_{app}), ion mobility (μ_0), and nanowire spacing (d) are all inversely related to the total time (τ^{tot}), meaning that doubling the applied voltage, selecting a material with twice the ion mobility, and doubling the spacing between

nanowires, for instance, would each independently reduce the total time required for filament growth by a factor of two, making it grow twice as fast. It does seem odd that increasing d would result in faster total growth, so this unexpected result should be further explored to confirm that the nondimensionalization was properly performed. For the remainder of this discussion, this peculiarity is disregarded but should be returned to at a later time for further justification.

4.1.3 Computational Validation Using Realistic System Parameters

The nondimensionalized relationships extracted from the simulation should be validated numerically using parameters that describe a realistic system. Thus, quantities describing a potential experimental configuration are employed as justification for the nondimensionalized model. The relationships between variables generated using the nondimensionalized case should also be observed in the case of particular single two-nanowire junctions in a dielectric with known dimensions, properties, and geometries.

As an example, a system consisting of two perpendicular silver (Ag) nanowires separated by a distance, d, of 50 nm is considered with the same curved-tip geometry as the nondimensionalized case. Initial conditions are made to match those of this initial system where the initial filament length is set equal to 10% of d (5 nm), and the radius is set to 5% of d (2.5 nm). This relative ratio between filament dimension and nanowire spacing is supported by the literature.[17] The rightmost nanowire experiences an applied voltage of V_{app} = 1.2 V with the addition of an electric potential. This applied

voltage of ~ 1 V is suitable for initiating the formation of volatile resistive memory.[7,18] Specifically, the application of ~ 1.2 V has been demonstrated to yield volatile resistive memory in an Ag-nanowire network within an amorphous polymeric electrolyte.[26]

The nanowires are assumed to be in an amorphous SiO_2 medium such that the Ag ion mobility μ_0 can be calculated from theoretical equations[27] and known physical parameters.[28] The ion hopping mechanism describing carrier drift in highly defective materials, such as amorphous solids or polymers, was used to make this calculation. In this model, the carrier drift velocity under low electric fields can be approximated to the following equation, **Equation 10**,[27] where v_i is the ion drift velocity, v is the jump attempting frequency, r is the jumping distance, e is the elemental charge, k_B is the Boltzman constant, T is temperature (assumed to be room temperature in this case), E is applied electric field (equivalent to the gradient of the electric potential, $\nabla\Phi$) , and ΔG is the energy barrier for ion hopping to occur. These physical parameters can be found in the literature.[28]

$$v_i \sim \left(\nu r^2 \frac{e}{k_B T}\right) E exp\left(-\frac{\Delta G^{\neq}}{k_B T}\right) \tag{10}$$

Based on **Equation 10**, the ion mobility μ_0 of Ag in amorphous SiO_2 is calculated to be $3.178 \cdot 10^{-11}$ $m^2/(V \cdot s)$.

The known quantities of d, μ_0, and V_{app} (50 nm, $3.178 \cdot 10^{-11}$ $m^2/(V \cdot s)$, and 1.2 V, respectively) are plugged into **Equation 9** to form **Equation 11** which can be used to calculate the time, $\tau = \tau^{tot}$, needed for the filament to be one time step away from

spanning the distance, d, separating the two nanowires. This miniscule separation prevents the calculations from undershooting the distance, d, since the simulation does not capture when exactly the system reaches τ^{tot} but rather finds the time at which the total nondimensionalized filament length exceeds 1. Given the small time step Δt = 0.0001, this effect should be negligible.

$$x = 2.96714 \cdot 10^{-9} exp\left(16499.3\sqrt{\tau} \right) + 1.43389 \cdot 10^{-9} \tag{11}$$

Constants in this nonlinear exponential **Equation 11** were found to be a = $2.96714 \cdot 10^{-9}$ m, b = 16499.3 $s^{-1/2}$, and c = $1.43389 \cdot 10^{-9}$ m. **Equation 11** is plotted in **Figure 4** below. By plugging x = d into **Equation 11**, $\tau = \tau^{tot}$ is determined to be $2.76931 \cdot 10^{-8}$ s , or approximately 28 ns. This value seems reasonable based on similar examined systems that have values of τ^{tot} on the order of hundreds of nanoseconds, with some variation.[17,18]

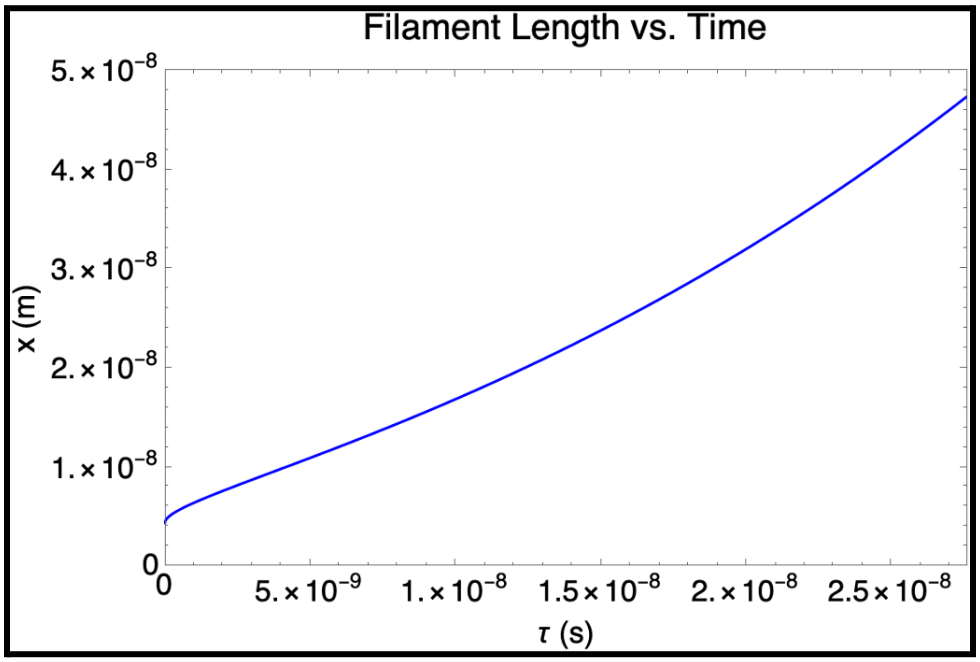


Figure 4: Predictive curve of filament growth rate for potential experimental configuration. The system consists of two Ag nanowires separated by d = 50 nm in an SiO_2 medium such that $\mu_0 = 3.178 \cdot 10^{-11}$ $m^2/(V \cdot s)$ and undergoes an applied voltage of $V_{app} = 1.2$ V. The filament has an initial length of 5 nm (10% of d) and a radius of 2.5 nm (5% of d) as with the nondimensionalized case. This curve is described by **Equation 11**. Note that this particular system matches the nondimensionalized case in **Figure 3**.

As with the nondimensionalized case, several parameter relationships can be evaluated. Looking to the values in **Figure 4** and using the conversion between τ^{non} and τ from **Equation 5**, the applied voltage (V_{app}), ion mobility (μ_0), and nanowire spacing (d) in the potential experimental configuration are all confirmed to be inversely related to the total time, τ^{tot}. Thus, doubling the applied voltage, selecting a material with twice the ion mobility, and doubling the spacing between nanowires, were confirmed to each independently reduce the total time required for filament growth by a factor of two from $\tau^{tot} = 2.7606 \cdot 10^{-8}$ s to $\tau^{tot} = 1.3803 \cdot 10^{-8}$ s, making it grow twice as fast. This confirms that the model presented in this work can be utilized in the design of any such nanowire system to yield physically relevant filament growth predictions.

4.1.4 Effect of Changing Initial Filament Dimensions

Since the computational model requires the input of initial filament length and radius that vary from system to system, their effects should be considered. Increasing the initial filament length does not affect the quantitative results of the simulation but merely truncates the generated data points such that they begin at the coordinate associated with the filament tip position. Changing the filament radius, however, impacts $\tau^{non,tot}$ and is worth modeling. The nondimensionalized effect of filament radius on growth rate is shown in **Figure 5**.

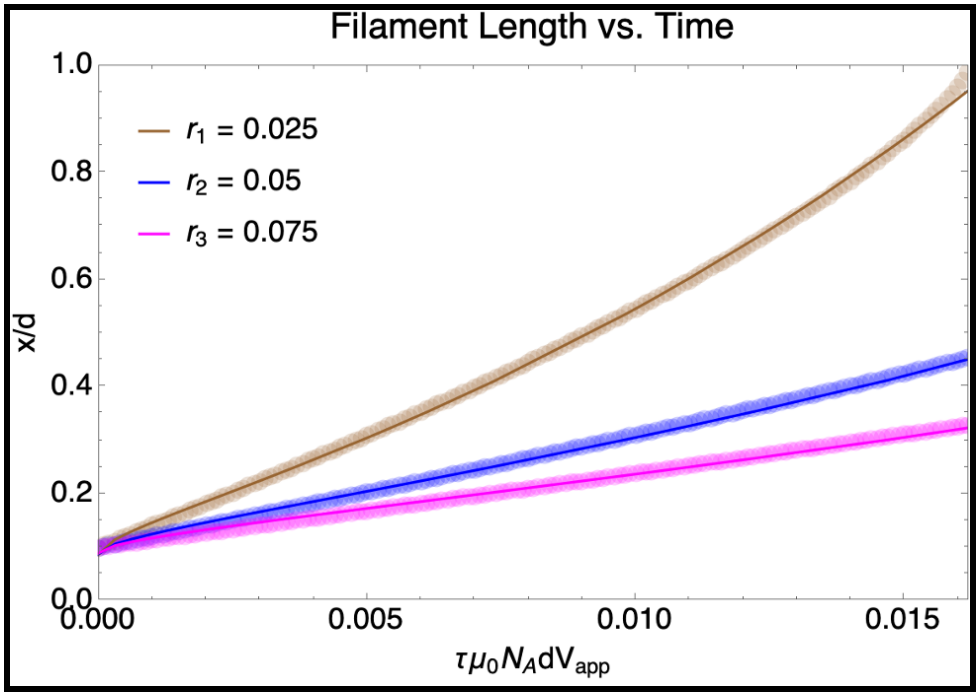


Figure 5: Effect of filament radius on growth rate. The nondimensionalized filament radius is set as r_1 = 0.025 (brown top curve), r_2 = 0.05 (blue middle curve), and r_3 = 0.075 (bottom magenta curve) for three respective simulations. The thick, half-opacity curves are composed of computationally-generated data points. The thin, full-opacity curves fit this data. The resulting curves are shown here.

The radius-dependent values of $\tau^{non,tot}$ were extracted from **Figure 5** and compared. $\tau^{non,tot}$ was found to be 0.0163, 0.0318, and 0.047 for r_1 = 0.025, r_2 = 0.05, and r_3 = 0.075, respectively. Since the radius was increased by a factor of two from r_1 to r_2 and by a factor of three from r_1 to r_3, the corresponding ratios of $\tau^{non,tot}$ were calculated. The ratio between $\tau^{non,tot}$ of r_2 and r_1 was 0.0318 / 0.0163, or 1.95092, close to the factor of 2 increase. The ratio between $\tau^{non,tot}$ of r_3 and r_1 was 0.047 / 0.0163, or 2.88344, close to the factor of 3 increase. These results indicate that there is a direct linear relationship between the filament radius and the time required for the filament to span d, meaning that increasing the radius by a factor of two, for instance, will increase the corresponding τ^{tot} by just under that same factor. However, the slight undershooting illustrates the minor effect of having a curved filament tip geometry (i.e.

not perfectly rectangular). Thus, increasing the radius by greater factors would cause the increase in $\tau^{non,tot}$ to approach that factor due to decreasing effects of the filament tip geometry.

Conceptually, a thinner filament requires the addition of fewer particles, and therefore less flux, to grow longer. This is because, in the case of a simple rectangular filament geometry, the filament area is defined as the product of the filament length and the width. Since the flux is dictated by the electric potential, which is the same in all three conditions illustrated in **Figure 5**, the "area in" is also consistent between conditions. Since the width (or radius) is also fixed, the length is the only dynamically updating dimension. Therefore, for thinner filaments, the "area in" quantity has a greater effect on length, causing the length to increase more rapidly for thinner filaments. Equivalently, the larger the filament radius, the slower the filament will grow.

4.2 Current as a Function of Time

4.2.1 Determination of Governing Equation for Time-Dependent Current

Tabulated filament growth simulation results can additionally be used to extract a relationship between electric current and time through nonlinear data fitting methods in Mathematica. However, the relationship that describes current and time does not fit into a typical nonlinear form, so a more manual manipulation of possible equation parameters is needed to find a reasonable fit. As with the original simulations, the initial filament length is assumed to be 0.1, and the radius is assumed to be 0.05. Assessing

flux as a function of time is analogous to assessing current versus time in the nondimensionalized case since flux describes the influx of cations, which is equivalent to the current ("area in"). The relationship between electric current and time for this case (with initial filament length of 0.1 and width of 0.05) has been found to be described by **Equation 12**.

$$I = (-5.58367 \cdot 10^{8} exp\left(-3.48577 \cdot 10^{9} \left(\mu_0 N_A dV_{app} \tau \right)^{14.6417} \right) + 0.00205608 \mu_0 N_A dV_{app} \tau + 5.58367 \cdot 10^{8}) \mu_0 N_A dV_{app} \quad (12)$$

Constants in this nonlinear exponential equation were found to be a = -5.58367, b = $-3.48577 \cdot 10^{9}$, c = 14.6417, d = 0.00205608, e = $5.58367 \cdot 10^{8}$. Note that the d in this case is just a constant equation parameter rather than equivalent to the spacing between two nanowires. **Equation 12** has an adjusted R^2 value of 0.996492 and parameter confidence intervals[25] with a confidence level of 95% for a, b, c, d, and e, respectively, as follows: ($-5.58367 \cdot 10^{8}$, $-5.58367 \cdot 10^{8}$), ($-3.48577 \cdot 10^{9}$, $-3.48577 \cdot 10^{9}$), (14.6327,14.6508), ($5.58367 \cdot 10^{8}$, $5.58367 \cdot 10^{8}$), and (0.00192471, 0.00218744). This means that the true parameters for a, b, c, d, and e are 95% likely to lie within the specified ranges. **Equation 12** is plotted in **Figure 6** along with the relevant data. Because the nondimensionalized time step $\Delta t = 0.0001$ is so small, it may be difficult to distinguish between the different data points. The thick, half-opacity blue curve consists of these data points, and the thin, full-opacity blue curve that overlaps with it is **Equation 12**.

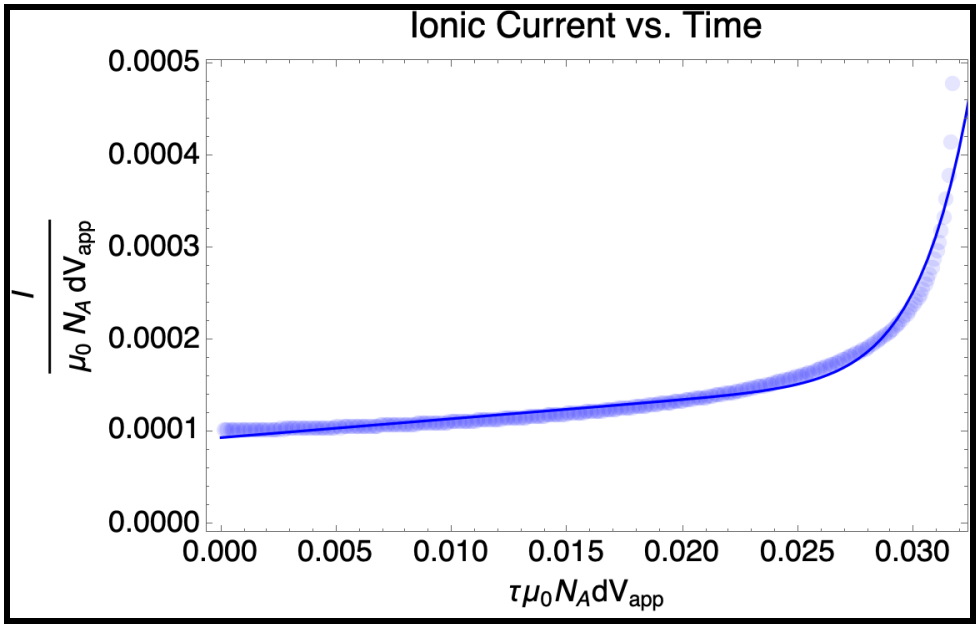


Figure 6: Fitted curve of nondimensionalized electric current versus time. The filament has an initial length of 0.1 and a radius of 0.05. The thick, half-opacity blue curve is composed of computationally-generated data points. The thin, full-opacity blue curve fits this data and is described by **Equation 12**. Note the nonlinear exponential relationship.

Using either **Equation 12** or the fitted curve in **Figure 6** allows for the prediction of electric current in the system as a function of time. This relationship appears to be nonlinearly exponential, with parameters closely dependent on one another. Contrary to their effects on total time (τ^{tot}), the applied voltage (V_{app}), ion mobility (μ_0), and nanowire spacing (d) are all directly related to the maximum ionic current ($I^{max,ionic}$) and electric current ($I^{max,electric}$), meaning that doubling the applied voltage, selecting a material with twice the ion mobility, and doubling the spacing between nanowires, for instance, would each independently increase the maximum ionic current by a factor of two.

4.2.2 Computational Validation Using Realistic System Parameters

As with the comparison between filament length and time, nondimensionalized relationships extracted from the simulation should be validated numerically using parameters that describe a realistic system. As with the prior computational validation, a potential experimental configuration is employed as justification for the nondimensionalized model which matches the configuration used previously.

The known quantities of d, μ_0, and V_{app} (50 nm, $3.178 \cdot 10^{-11}$ $m^2/(V \cdot s)$, and 1.2 V, respectively) are plugged into **Equation 12** to form **Equation 13** and **Equation 14** which can be used to calculate the final, maximum current, $I = I^{max}$. The quantity I^{max} is reached when the filament is one time step away from spanning the distance, d, separating the two nanowires. **Equation 13** describes the ionic current, I_{ionic}, in units of ions/s and can be multiplied by the fundamental positive charge, $e = 1.602 \cdot 10^{-19}$ C to get to **Equation 14**. **Equation 14** describes the electric current, $I_{electric}$, in units of amperes (A).

$$I_{ionic} = (-5.58367 \cdot 10^8 exp\left(-1.29616 \cdot 10^{97} \tau^{14.6417} \right) + 1967.5\tau + 5.58367 \cdot 10^8) \, 956918 \tag{13}$$

$$I_{electric} = (-5.58367 \cdot 10^8 exp\left(-1.29616 \cdot 10^{97} \tau^{1.46417} \right) + 1.9675 \cdot 10^3 \tau + 5.58367 \cdot 10^8) \left(1.53298 \cdot 10^{-13} \right) \tag{14}$$

Constants in this nonlinear exponential **Equation 13** were found to be a = $-5.58367 \cdot 10^8$, b = $-1.29616 \cdot 10^{97}$, c = 14.6417, d = 1967.5, e = $5.58367 \cdot 10^8$, and f = 956918 with the appropriate units to yield units of ions/s. Similarly, constants in

Equation 14 were found to be a = -5.58367·10^{8}, b = -1.29616·10^{97} s$^{-1/2}$, c = 1.46417, d = 1967.5, e = 5.58367·10^{8}, and f = 1.53298·10^{-13} with the appropriate units to yield units of Amperes. **Equation 13** is plotted in **Figure 7** below. As before, plugging x = d into **Equation 11**, yields τ^{tot} = 2.7606·10^{-8} s , or approximately 33 ns, and seems reasonable based on the literature.[17] By plugging τ^{tot} = 2.7606·10^{-8} s into **Equation 13** and **Equation 14**, respectively, the maximum ionic current and maximum electric current are determined to be I^{max,ionic} = 434.723 ions/s and I^{max,electric} = 6.96426·10^{-17}A.

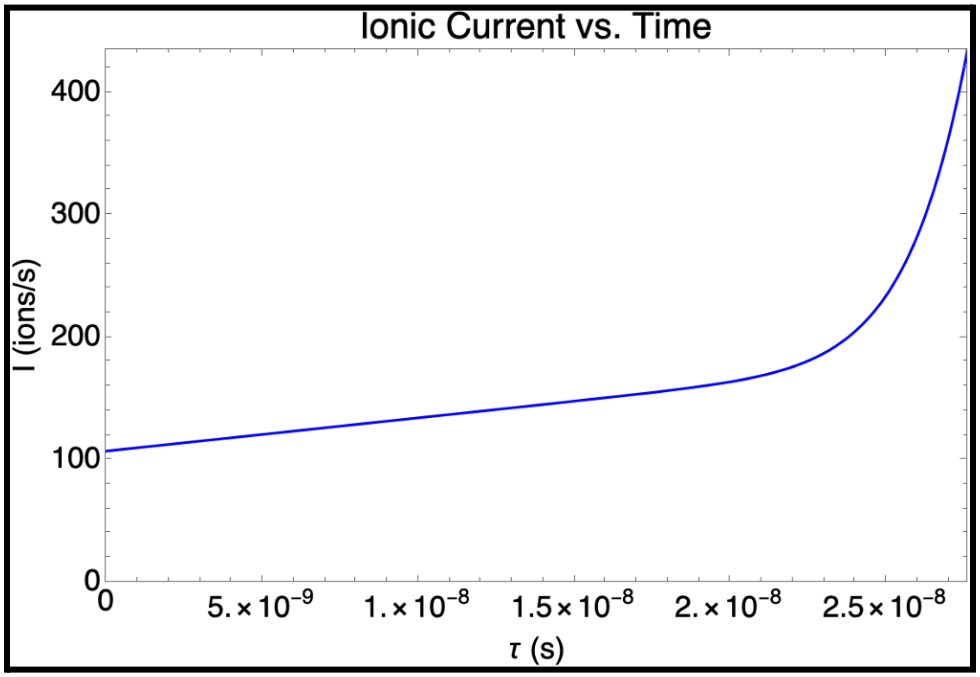


Figure 7: Predictive curve of ionic current versus time for potential experimental configuration. The system consists of two Ag nanowires separated by d = 50 nm in an SiO_2 medium such that μ_0 = 3.178·10^{-11} m^2/(V·s) and undergoes an applied voltage of V_{app} = 1.2 V. The filament has an initial length of 5 nm (10% of d) and a radius of 2.5 nm (5% of d) as with the nondimensionalized case. This curve is described by **Equation 13**. Note that this particular system matches the nondimensionalized case in **Figure 6**.

As with the nondimensionalized case, several parameter relationships can be evaluated. Plugging τ^{tot} = 2.7606·10^{-8} from previous calculations into **Equation 13** and

using **Equation 6**, the applied voltage (V_{app}), ion mobility (μ_0), and nanowire spacing (d) in the potential experimental configuration are all confirmed to be inversely related to the total time, τ^{tot}. Thus, doubling the applied voltage, selecting a material with twice the ion mobility, and doubling the spacing between nanowires, were confirmed to each independently increase the maximum ionic current and the maximum electric current during filament growth by a factor of two from $I^{max,ionic}$ = 434.723 ions/s to $I^{max,ionic}$ = 869.446 ions/s and from $I^{max,ionic} = 6.96426 \cdot 10^{-17}$ A to $I^{max,ionic} = 1.39285 \cdot 10^{-17}$ A, respectively. This confirms that the model presented in this work can be utilized in the design of any such nanowire system to yield physically relevant predictions of current. In lieu of reassessing the effect of changing initial filament dimensions on current, the results of **Figure 5** can be reasonably extended to represent the roughly linear inverse dependence of the final maximum current on the filament radius.

5. CONCLUSIONS AND FUTURE DIRECTIONS

This work presents a versatile computational model and resulting governing equations for predicting and understanding filament formation and growth behavior in the presence of an applied electric potential. This analysis specifically focuses on uncovering the relationships between both filament growth and time, and current and time, from a small initial filament geometry until the time when the electric potential source is reached. A nondimensionalization approach facilitates the development of a widely-applicable model that is independent of fundamental system parameters. Thus,

the resulting model can be tailored to represent any two-nanowire junction under any applied voltage. This allows for the input of system-specific geometric parameters such as filament width, initial filament length, and nanowire spacing, as well as materials-specific parameters such as ion mobility (μ) and ion dimension during the application of a voltage of interest.

This work shows promise for aiding in the design of physical, brain-like systems for enhanced computing efficiency. Additionally, due to the vast customizability of the simulation, this tool can be simply adapted for studying filament formation in distinct application contexts that extend beyond computation. For instance, predicting filament formation in both liquid-electrolyte and solid-state batteries, for instance, is of particular relevance as filament growth remains batteries' primary failure mode, causing short circuits.[29]

To enhance the model's accuracy and reliability, it is imperative to reduce simplifications and reassess underlying assumptions. For instance, considering presently disregarded mechanisms such as ion diffusion, redox reactions, and tunneling current as well as surface energy effects and the nonlinear relationship between electric field and flux will elevate the model to become a more comprehensive representation of the underlying dynamics. Further expansion of the scope could include considering the filament growth and resulting current after the filament spans the nanowire spacing as well as the effects of removing and re-initiating the applied voltage such that volatile, and in the case of higher applied voltages, non-volatile,

resistive switching could be observed. Additional consideration of the resistance of silver (Ag) nanowire junction could assist in determining current behavior once the filament spans the nanowire spacing. [31]

Additionally, refining the model to include a filament geometry that more closely resembles dendritic structures present in neuron growth can offer a deeper understanding of the intricate growth mechanisms. Another interesting area of exploration involves examining how changes in filament geometry, such as surface curvature, impact the overall filament growth behavior.

Furthermore, this model can be expanded to represent multiple nanowire junctions in the context of a nanowire network. Nanowire network and tunneling effects in 3D systems can be computationally modeled and physically fabricated for experimental validation.[26,30] **Figure 8** showcases a physically fabricated early-stage nanowire network model that serves as a preliminary demonstration of the potential for future expansion of this work.

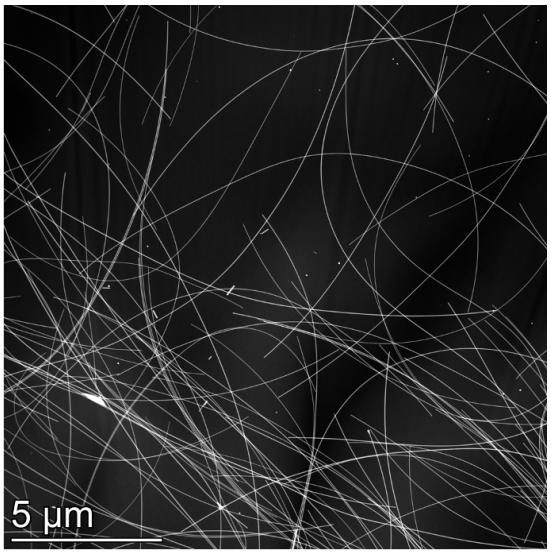


Figure 8: HR-TEM image of Ag nanowire network. To prepare the specimen, a silver (Ag) nanowire solution (0.833 mg/ml of silver nanowire dispersed in ethyl alcohol) was twice spin-coated on a copper grid at 1000 rpm for 30 s. The copper grid and nanowire network system was then vacuum annealed at 80 deg C overnight. The HR-TEM analysis was conducted by Dr. Jin-Hoon Kim using the Titan Themis Z G3 at MIT Nano. Scale bar: 5 μm.

These future endeavors will undoubtedly pave the way for enhanced insights and advancements in the understanding and application of the proposed model. Overall, this work serves as a fundamental first step towards predicting filament growth behavior and the accompanying current profiles, enabling a methodical exploration of diverse system dynamics.

REFERENCES

1. Herculano-Houzel S. The human brain in numbers: a linearly scaled-up primate brain. *Front Hum Neurosci*. 2009;3. doi:10.3389/neuro.09.031.2009
2. Schuman CD, Kulkarni SR, Parsa M, Mitchell JP, Date P, Kay B. Opportunities for neuromorphic computing algorithms and applications. *Nat Comput Sci*. 2022;2(1):10-19. doi:10.1038/s43588-021-00184-y
3. Bassett DS, Gazzaniga MS. Understanding complexity in the human brain. *Trends Cogn Sci*. 2011;15(5):200-209. doi:10.1016/j.tics.2011.03.006
4. Hochstetter J, Zhu R, Loeffler A, Diaz-Alvarez A, Nakayama T, Kuncic Z. Avalanches and edge-of-chaos learning in neuromorphic nanowire networks. *Nat Commun*. 2021;12(1):4008. doi:10.1038/s41467-021-24260-z
5. Ou W, Xiao S, Zhu C, Han W, Zhang Q. An overview of brain-like computing: Architecture, applications, and future trends. *Front Neurorobotics*. 2022;16:1041108. doi:10.3389/fnbot.2022.1041108
6. Martins NRB, Angelica A, Chakravarthy K, et al. Human Brain/Cloud Interface. *Front Neurosci*. 2019;13:112. doi:10.3389/fnins.2019.00112
7. Wang Z, Joshi S, Savel'ev SE, et al. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat Mater*. 2017;16(1):101-108. doi:10.1038/nmat4756
8. Südhof TC. The cell biology of synapse formation. *J Cell Biol*. 2021;220(7):e202103052. doi:10.1083/jcb.202103052
9. Jabeen S, Thirumalai V. The interplay between electrical and chemical synaptogenesis. *J Neurophysiol*. 2018;120(4):1914-1922. doi:10.1152/jn.00398.2018
10. Schwierz F, Pezoldt J, Granzner R. Two-dimensional materials and their prospects in transistor electronics. *Nanoscale*. 2015;7(18):8261-8283. doi:10.1039/C5NR01052G
11. Stephan Menzel, Kaupmann P, Waser R. Understanding filamentary growth in electrochemical metallization memory cells using kinetic Monte Carlo simulations. *Nanoscale*. 2015;7(29):12673-12681. doi:10.1039/C5NR02258D
12. Loeffler A, Diaz-Alvarez A, Zhu R, et al. Neuromorphic learning, working memory, and metaplasticity in nanowire networks. *Sci Adv*. 2023;9(16):eadg3289. doi:10.1126/sciadv.adg3289
13. Wang W, Laudato M, Ambrosi E, et al. Volatile Resistive Switching Memory Based on Ag Ion Drift/Diffusion Part I: Numerical Modeling. *IEEE Trans Electron Devices*. 2019;66(9):3795-3801. doi:10.1109/TED.2019.2928890
14. Wang W, Laudato M, Ambrosi E, et al. Volatile Resistive Switching Memory Based on Ag Ion Drift/Diffusion—Part II: Compact Modeling. *IEEE Trans Electron Devices*. 2019;66(9):3802-3808. doi:10.1109/TED.2019.2928888
15. Zahoor F, Azni Zulkifli TZ, Khanday FA. Resistive Random Access Memory (RRAM): an Overview of Materials, Switching Mechanism, Performance, Multilevel Cell (mlc) Storage, Modeling, and Applications. *Nanoscale Res Lett*. 2020;15(1):90.

doi:10.1186/s11671-020-03299-9
16. Lv S, Wang H, Zhang J, Liu J, Sun L, Yu Z. An Analytical Model for the Forming Process of Conductive-Bridge Resistive-Switching Random-Access Memory. *IEEE Trans Electron Devices*. 2014;61(9):3166-3171. doi:10.1109/TED.2014.2341274
17. Menzel S, Tappertzhofen S, Waser R, Valov I. Switching kinetics of electrochemical metallization memory cells. *Phys Chem Chem Phys*. 2013;15(18):6945. doi:10.1039/c3cp50738f
18. Fu T, Liu X, Gao H, et al. Bioinspired bio-voltage memristors. *Nat Commun*. 2020;11(1):1861. doi:10.1038/s41467-020-15759-y
19. Safra JE. Avogadro's Number. In: *The New Encyclopaedia Britannica*. 15th ed. Encyclopaedia britannica; 2007.
20. Ellingson, Steven W. *Electromagnetics*. 1st ed. VT Publishing; 2018. doi:10.21061/electromagnetics-vol-1
21. Hazewinkel M. Laplace equation. In: *Encyclopaedia of Mathematics*. Springer-Verlag; 2002.
22. Walet NR. 3.2: Explicit Boundary Conditions. In: *Partial Differential Equations*. ; 2010. Accessed May 18, 2023. https://oer.physics.manchester.ac.uk/PDEs/Notes/jsmath/Notes.html
23. Crawford M. 3.2: Diffusion-Controlled Reactions. In: *Chem 322: Physical Chemistry II*. ; 2023. https://chem.libretexts.org/Courses/Knox_College/Chem_322%3A_Physical_Chemisty_II
24. Yang H, Wu N. Ionic conductivity and ion transport mechanisms of solid-state lithium-ion battery electrolytes: A review. *Energy Sci Eng*. 2022;10(5):1643-1671. doi:10.1002/ese3.1163
25. Hazra A. Using the confidence interval confidently. *J Thorac Dis*. 2017;9(10):4124-4129. doi:10.21037/jtd.2017.09.14
26. Zhu Y, Chen J, Wan T, et al. Convertible Insulator–Conductor Transition in Silver Nanowire Networks: Engineering the Nanowire Junctions. *ACS Appl Electron Mater*. 2019;1(7):1275-1281. doi:10.1021/acsaelm.9b00218
27. Valov I, Waser R, Jameson JR, Kozicki MN. Electrochemical metallization memories—fundamentals, applications, prospects. *Nanotechnology*. 2011;22(25):254003. doi:10.1088/0957-4484/22/25/254003
28. Lübben M, Menzel S, Park SG, Yang M, Waser R, Valov I. SET kinetics of electrochemical metallization cells: influence of counter-electrodes in SiO_2/Ag based systems. *Nanotechnology*. 2017;28(13):135205. doi:10.1088/1361-6528/aa5e59
29. Liu X, Garcia-Mendez R, Lupini AR, et al. Local electronic structure variation resulting in Li 'filament' formation within solid electrolytes. *Nat Mater*. 2021;20(11):1485-1490. doi:10.1038/s41563-021-01019-x
30. Chopin C, De Wergifosse S, Marchal N, Van Velthem P, Piraux L, Abreu Araujo F. Memristive and Tunneling Effects in 3D Interconnected Silver Nanowires. *ACS*

Omega. 2023;8(7):6663-6668. doi:10.1021/acsomega.2c07171
31. Selzer F, Floresca C, Kneppe D, et al. Electrical limit of silver nanowire electrodes: Direct measurement of the nanowire junction resistance. *Appl Phys Lett*. 2016;108(16):163302. doi:10.1063/1.4947285

APPENDIX

Modeling Synapse Formation and Growth as a Non-Biological Analog to the Brain

The undergraduate thesis of Sara V. Fernandez as submitted to the Department of Materials Science and Engineering at the Massachusetts Institute of Technology, May 23, 2023

Building the Model's Initial State

Filament Geometry

Package-Loading

In[ ]:=
```
Needs["NDSolve`FEM`"]
Needs["DifferentialEquations`NDSolveUtilities`"];
Needs["DifferentialEquations`InterpolatingFunctionAnatomy`"];
```

Inputting Initial Geometry Values

In[ ]:=
```
n = 12; (* n is related to the degree of mesh refinement.*)
initialFilamentLength = 0.1;
radius = 0.05; (*This was changed to include 0.025, 0.05,
and 0.075 to assess the effect of changing initial filament
 dimensions and for the production of the Supplementary Videos.*)
```

Radius

This equation is used to construct the quarter-circular filament tip geometry. Solving for the initialFilamentAreaExpression allows for the radius to be used as an input.

```
initialFilamentArea = (initialFilamentAreaExpression /.
    {Flatten[Solve[radius == 1/(4 - π) (2 initialFilamentLength - √2
                √(2 initialFilamentLength^2 + 1/2 (-8 + 2 π) initialFilamentAreaExpression)),
      initialFilamentAreaExpression]]})[[1]]
```

Filament Tip Geometry

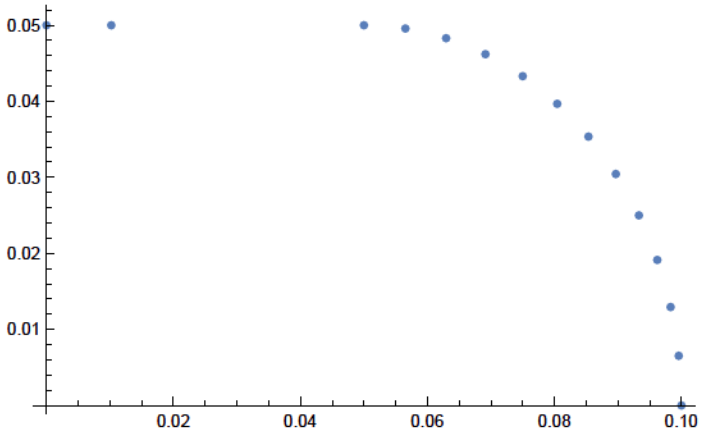
This function was adapted with permission from a similar function created by Dr. W. Craig Carter.

```
In[•]:= tipCoords[initialFilamentLength_, initialFilamentArea_, n_] := Block[

  (*The following describes the geometry of the quarter-circle tip.*)
  {radius = 1/(4 - π) (2 initialFilamentLength -
       √2 √(2 initialFilamentLength^2 + 1/2 (-8 + 2 π) initialFilamentArea)),
   flankPoints
   },

  (*The following describes the geometry of the points that flank the quarter-
   circle tip (i.e. the horizontally-translated points).*)
  If[initialFilamentLength - 1.5 radius ≥ 0,
   flankPoints = Sqrt[(initialFilamentLength - radius)]
     Sqrt[Range[0, initialFilamentLength - 1.5 radius,
        (initialFilamentLength - 1.5 radius)] / n], flankPoints = {0}];
  Join[{#, radius} & /@ flankPoints,
   Threaded[{initialFilamentLength - radius, 0}] +
    radius AngleVector /@ Range[Pi / 2, 0, -Pi/(2 n)]
  ]
 ]
```

Output Example: Visualization

In[ ]:=
```
tipCoordsVisualization[
  initialFilamentLength_, initialFilamentArea_, n_] := ListPlot[
  tipCoords[initialFilamentLength, initialFilamentArea, n], PlotRange → Full]
```

In[ ]:=
```
tipCoordsVisualization[initialFilamentLength, initialFilamentArea, n];
```



Mesh Construction

The mesh construction functions were adapted with permission from similar function created by Dr. W. Craig Carter.

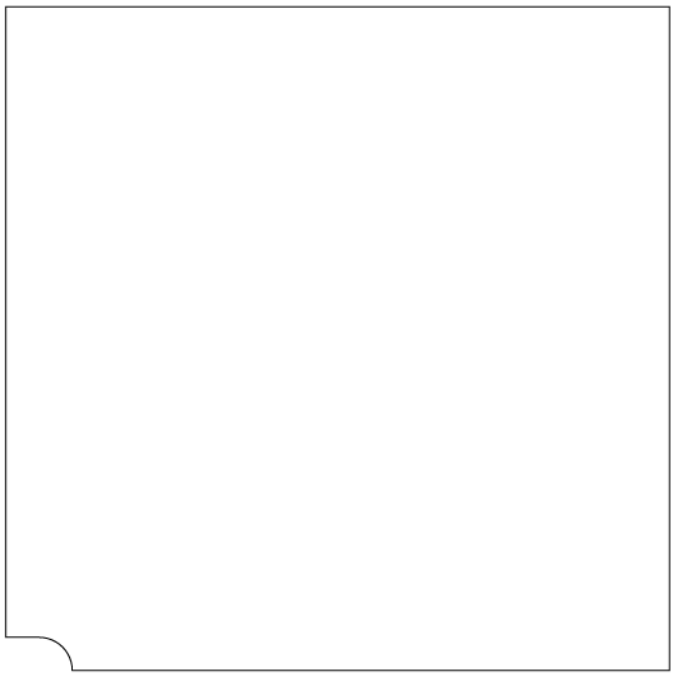
Boundary Mesh

```
In[ ]:= makingBoundaryMesh[initialFilamentLength_, initialFilamentArea_, n_] :=
 Block[{height = 1,
   flank = tipCoords[initialFilamentLength, initialFilamentArea, n],
   coords,
   flankPointCount,
   flankPointIndices,
   lines,
   boundaryMarkerFunction },
  flankPointCount = Length[flank];
  flankPointIndices = Range[flankPointCount];

  (*This draws lines between coordinates located around a 1x1 square to create
   an outline such that the filament geometry is shown using negative space.*)
  coords = Join[flank, {{1, 0}, {1, 1}, {0, 1}}];
  lines =
   Join[Most[Transpose[{flankPointIndices, RotateLeft[flankPointIndices]}]],
    {{flankPointCount, flankPointCount + 1},
     {flankPointCount + 1, flankPointCount + 2},
     {flankPointCount + 2, flankPointCount + 3},
     {flankPointCount + 3, 1}
    }];

  (*This creates a mesh of the outline.*)
  ToBoundaryMesh["Coordinates" → coords,
   "BoundaryElements" → {LineElement[lines
      , Join[ConstantArray[1, flankPointCount - 1], {2, 3, 4, 5}]]},
   "PointElements" →
    {PointElement[List /@ Range[Length[coords]], Range[Length[coords]]]}

  ]
 ]
```

Output Example: Visualization

```
In[ ]:= boundaryMeshVisualization[initialFilamentLength_, initialFilamentArea_, n_] := Show[
   makingBoundaryMesh[initialFilamentLength, initialFilamentArea, n]["Wireframe"]]
```

```
In[ ]:= boundaryMeshVisualization[initialFilamentLength, initialFilamentArea, n];
```



Area Mesh

In[•]:=
```
boundaryMesh[initialFilamentLength_, initialFilamentArea_, n_] :=
 makingBoundaryMesh[initialFilamentLength, initialFilamentArea, n]
```

```
makingAreaMesh[initialFilamentLength_, initialFilamentArea_, n_] :=
 With[{boundaryMesh =
    makingBoundaryMesh[initialFilamentLength, initialFilamentArea, n],
   radius = 1/(4 - π) (2 initialFilamentLength -
       √2 √(2 initialFilamentLength^2 + 1/2 (-8 + 2 π) initialFilamentArea))
  },
  <|"Boundary" → boundaryMesh,
   "Area" → ToElementMesh[boundaryMesh
     , MeshRefinementFunction →
      Function[{vertices, area}, Block[{xM, yM}, {xM, yM} = Mean[vertices];
        If[ Sqrt[(yM^2 + (xM - initialFilamentLength)^2)] < .1 ||
          (0 < xM < initialFilamentLength && 0 < yM < radius),
         area > 0.000001, area > 0.0001]
       ]
      ]
    ]
   |>
 ]
```

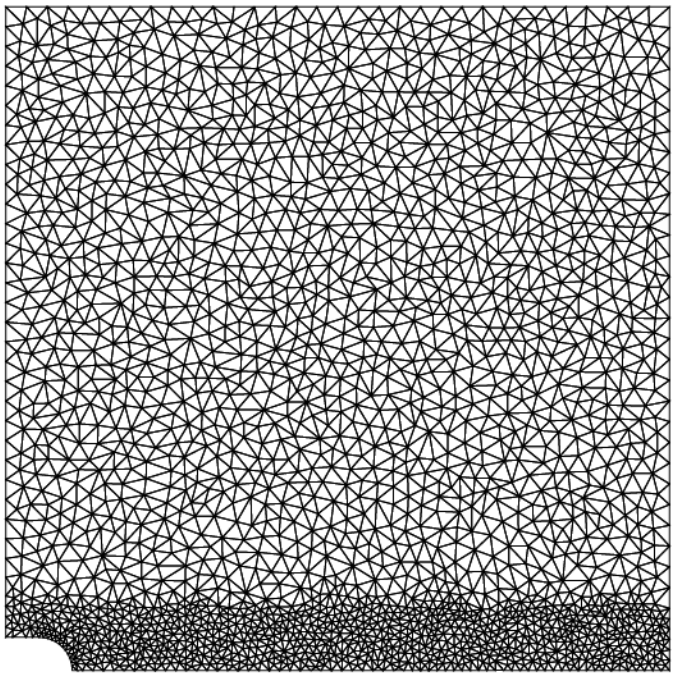
```
areaMesh[initialFilamentLength_, initialFilamentArea_, n_] :=
 makingAreaMesh[initialFilamentLength, initialFilamentArea, n]["Area"]
```

```
mesh[initialFilamentLength_, initialFilamentArea_, n_] :=
 mesh[initialFilamentLength, initialFilamentArea, n] = ToElementMesh[boundaryMesh[
    initialFilamentLength, initialFilamentArea, n], MeshRefinementFunction →
    Function[{vertices, area}, Block[{xM, yM}, {xM, yM} = Mean[vertices];
      If[ (Abs[yM] < .1 ), area > 0.0001, area > 0.0005]
     ]
    ]]
```

Output Example: Visualization

```
meshVisualization[initialFilamentLength_, initialFilamentArea_, n_] :=
  Show[mesh[initialFilamentLength, initialFilamentArea, n]["Wireframe"]];
```

```
meshVisualization[initialFilamentLength, initialFilamentArea, n];
```



Electric Potential

Magnitude and Position

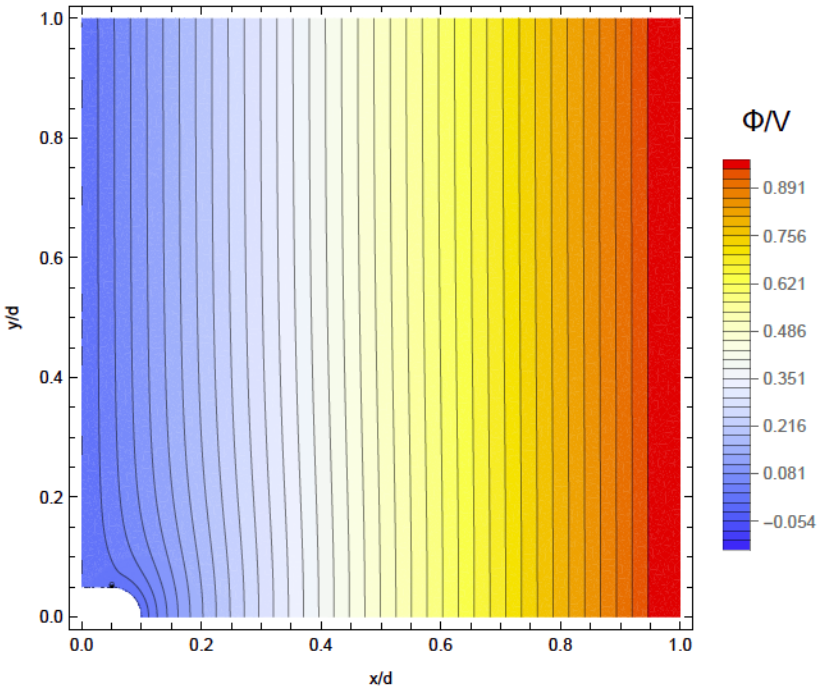
The numbers associated with the DirichletCondition functions indicate the magnitude of the applied voltage, or electric potential. The numbers associated with the ElementMarker functions indicate the position of the applied voltage where 1 is the curved filament tip, 2 is the remaining horizontal line that extends rightward from the filament tip, 3 is the rightmost vertical line, 4 is the horizontal line at the top that connects 1 and 5, and 5 is the leftmost vertical line.

In[ ]:=
```
electricPotential[initialFilamentLength_, initialFilamentArea_, n_] :=
 electricPotential[initialFilamentLength, initialFilamentArea, n] =
  NDSolveValue[
   {
    Laplacian[phi[x, y], {x, y}] ==
     NeumannValue[0, ElementMarker == 2] +
      NeumannValue[0, ElementMarker == 4],
    DirichletCondition[phi[x, y] == 1. , ElementMarker == 3 ],
    DirichletCondition[phi[x, y] == 0., ElementMarker == 1],
    DirichletCondition[phi[x, y] == 0., ElementMarker == 5]
   }
   , phi, {x, y} ∈ areaMesh[initialFilamentLength, initialFilamentArea, n]]
```

Output Example: Visualization

```
In[ ]:= electricPotentialVisualization[initialFilamentLength_, initialFilamentArea_, n_] :=
 electricPotentialVisualization[initialFilamentLength, initialFilamentArea, n] =
  Module[{localPhiSol = electricPotential[initialFilamentLength,
      initialFilamentArea, n]}, ContourPlot[localPhiSol[x, y],
    {x, y} ∈ areaMesh[initialFilamentLength, initialFilamentArea, n],
    ColorFunction → (ColorData["TemperatureMap"][#] &), Contours → 40, PlotLegends →
     BarLegend[Automatic, LegendLabel → "Φ/V"], FrameLabel → {"x/d", "y/d"}]]
```

```
In[ ]:= electricPotentialVisualization[initialFilamentLength, initialFilamentArea, n];
```



Note that there is a minor issue with using Mathematica's interpolation function on meshes, resulting in unexpected behavior at certain positions. The region directly surrounding (0.05, 0.05) shows this behavior. However, this issue is currently being patched in Mathematica and has no measurable effect on this model.

System Image

The visualization of the electric potential can be used to create an image of the entire modeled system by flipping the image of the half filament over its axis of symmetry.

Printed by Wolfram Mathematica Student Edition

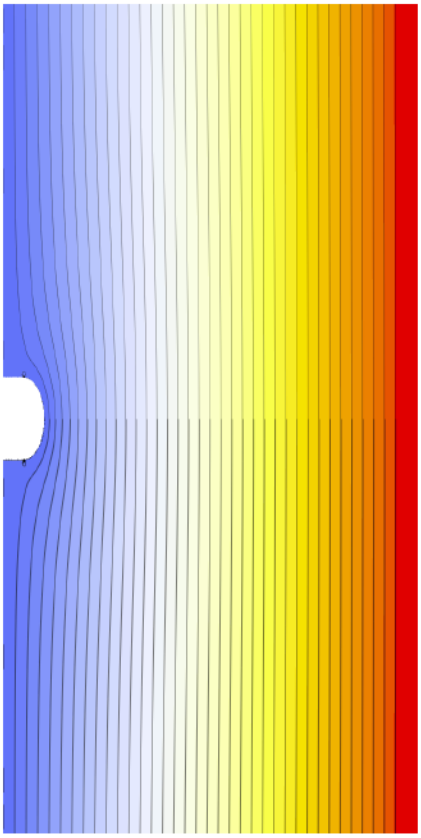
```
In[•]:= topHalfSystemImage[initialFilamentLength_, initialFilamentArea_, n_] :=
 topHalfSystemImage[initialFilamentLength, initialFilamentArea, n] =
  Module[{localPhiSol = electricPotential[initialFilamentLength,
      initialFilamentArea, n]}, ContourPlot[localPhiSol[x, y],
    {x, y} ∈ areaMesh[initialFilamentLength, initialFilamentArea, n],
    ColorFunction → "TemperatureMap", Contours → 40,
    Frame → False, PlotRange → {{0, 1}, {0, 0.5}}]]
```

```
In[•]:= bottomHalfSystemImage[initialFilamentLength_, initialFilamentArea_, n_] :=
 bottomHalfSystemImage[initialFilamentLength, initialFilamentArea, n] =
  ImageReflect[topHalfSystemImage[initialFilamentLength, initialFilamentArea, n]]
```

Output Example: Visualization

```
In[•]:= systemImageVisualization[initialFilamentLength_, initialFilamentArea_, n_] :=
 systemImageVisualization[initialFilamentLength,
   initialFilamentArea, n] = ImageAssemble[
   {{topHalfSystemImage[initialFilamentLength, initialFilamentArea, n]},
    {bottomHalfSystemImage[initialFilamentLength, initialFilamentArea, n]}},
   ImageSize → Medium]
```

```
In[•]:= systemImageVisualization[initialFilamentLength, initialFilamentArea, n];
```



Electric Field and Flux

Electric Field Components

Taking the derivative of the x and y components of the electric potential yields the electric field.

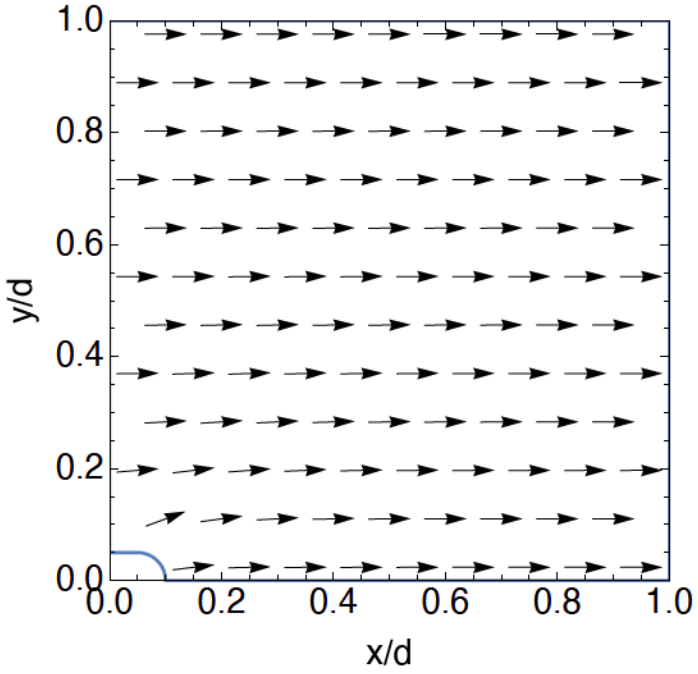
```
In[ ]:= dex[initialFilamentLength_, initialFilamentArea_, n_] :=
         dex[initialFilamentLength, initialFilamentArea, n] = Derivative[1, 0][
           electricPotential[initialFilamentLength, initialFilamentArea, n]]
        dey[initialFilamentLength_, initialFilamentArea_, n_] :=
         dey[initialFilamentLength, initialFilamentArea, n] = Derivative[0, 1][
           electricPotential[initialFilamentLength, initialFilamentArea, n]]
```

```
In[ ]:= electricField[initialFilamentLength_, initialFilamentArea_, n_] :=
         electricField[initialFilamentLength, initialFilamentArea, n] =
          Function[{x, y}, {dex[initialFilamentLength, initialFilamentArea, n][x, y],
            dey[initialFilamentLength, initialFilamentArea, n][x, y]}
          ]
```

Output Example: Visualization

```
In[ ]:= electricFieldVisualization[initialFilamentLength_, initialFilamentArea_, n_] :=
         electricFieldVisualization[initialFilamentLength, initialFilamentArea] =
          VectorPlot[electricField[initialFilamentLength, initialFilamentArea, n][x, y],
           {x, y} ∈ mesh[initialFilamentLength, initialFilamentArea, n],
           VectorPoints → 10, PlotRange → {{0, 1}, {0, 1}},
           PlotRangePadding → None, FrameLabel → {"x/d", "y/d"},
           LabelStyle → Directive[FontFamily → "Helvetica", Black, FontSize → 18],
           VectorColorFunction → (Black &), VectorScaling → None,
           VectorRange → All, RegionFillingStyle → None]
```

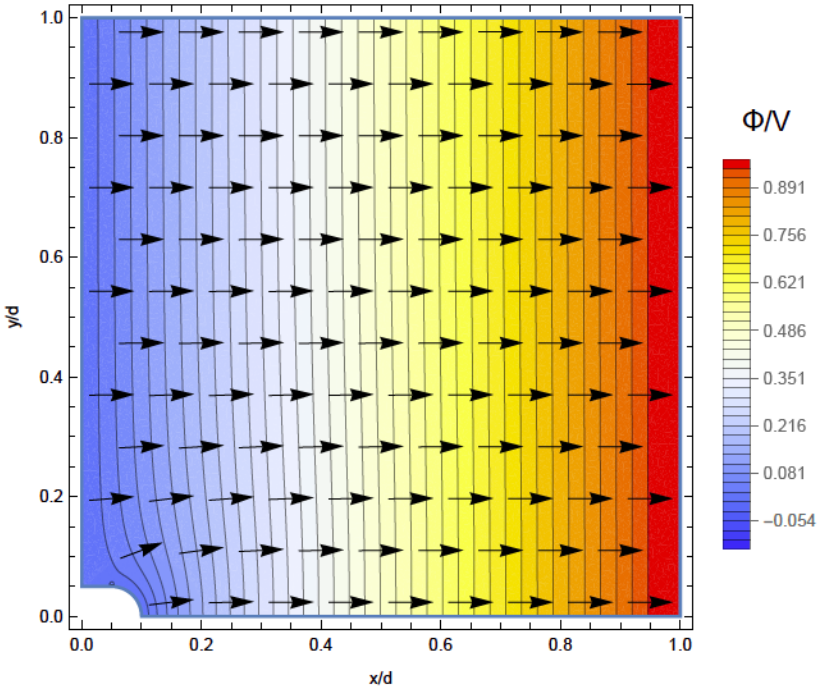
```
In[ ]:= electricFieldVisualization[initialFilamentLength, initialFilamentArea, n];
```



Here the electric potential and field visualization are overlaid.

In[ ]:=
```
electricPotentialAndFieldVisualization[initialFilamentLength_,
  initialFilamentArea_, n_] := electricPotentialAndFieldVisualization[
   initialFilamentLength, initialFilamentArea, n] = Show[
   electricPotentialVisualization[initialFilamentLength, initialFilamentArea, n],
   electricFieldVisualization[initialFilamentLength, initialFilamentArea, n]]
```

In[ ]:=
```
electricPotentialAndFieldVisualization[
  initialFilamentLength, initialFilamentArea, n];
```



Flux

In[ ]:=
```
mobility = 1;
```

The flux is considered to be equal to a constant mobility multiplied by the electric field. Mobility is treated as constant in this case due to several simplifying assumptions. The negative sign that is typically included in this equation is neglected in order to simplify computation since the one-dimensional direction of filament growth is known.

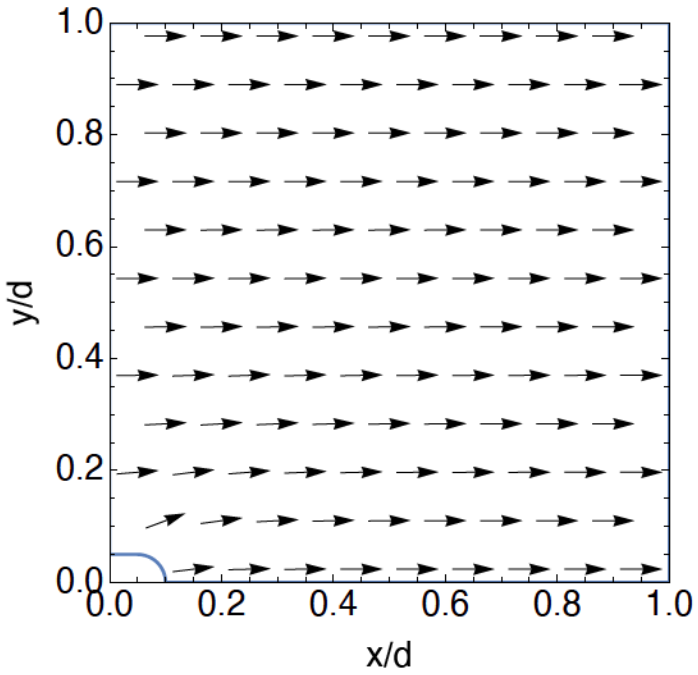
In[ ]:=
```
flux[initialFilamentLength_, initialFilamentArea_, n_, mobility_][x_, y_] :=
 flux[initialFilamentLength, initialFilamentArea, n, mobility][x, y] =
  mobility * electricField[initialFilamentLength, initialFilamentArea, n][x, y]
```

Output Example: Visualization

Since the flux is considered to be related to the electric field by a positive constant 1, the electric field and flux visualizations are identical for the nondimensionalized case, but both visualization types are shown for completeness. However, switching the sign from positive to negative in the flux equation can be done with ease.

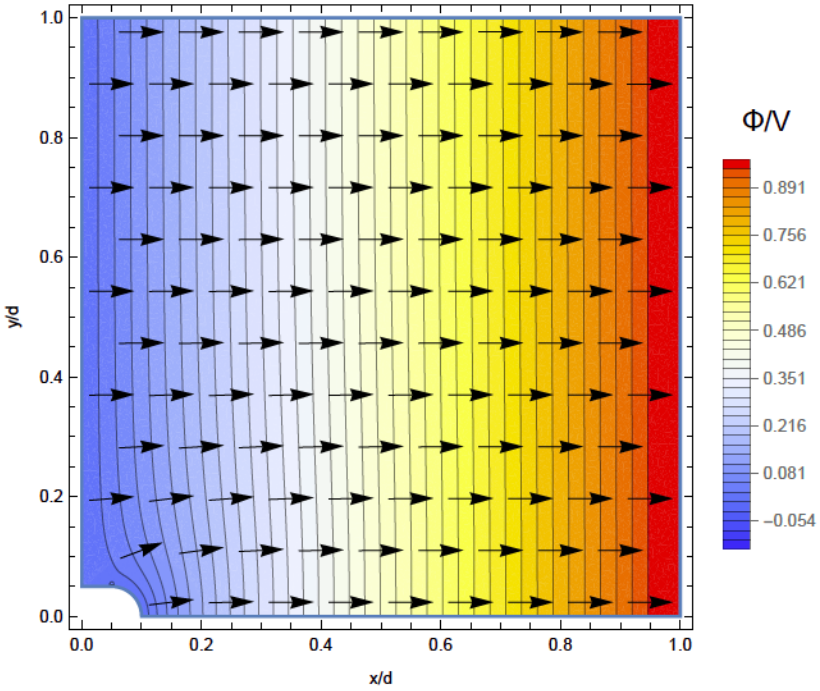
```
In[ ]:= fluxVisualization[initialFilamentLength_, initialFilamentArea_, n_, mobility_] :=
         fluxVisualization[initialFilamentLength, initialFilamentArea, n, mobility] =
          VectorPlot[flux[initialFilamentLength, initialFilamentArea, n, mobility][x, y],
           {x, y} ∈ mesh[initialFilamentLength, initialFilamentArea, n],
           VectorPoints → 10, PlotRange → {{0, 1}, {0, 1}},
           PlotRangePadding → None, FrameLabel → {"x/d", "y/d"},
           LabelStyle → Directive[FontFamily → "Helvetica", Black, FontSize → 18],
           VectorColorFunction → (Black &), VectorScaling → None,
           VectorRange → All, RegionFillingStyle → None]
```

```
In[ ]:= fluxVisualization[initialFilamentLength, initialFilamentArea, n, mobility];
```




```
In[ ]:= electricPotentialAndFluxVisualization[
          initialFilamentLength_, initialFilamentArea_, n_, mobility_] :=
         electricPotentialAndFluxVisualization[initialFilamentLength,
          initialFilamentArea, n, mobility] = Show[
          electricPotentialVisualization[initialFilamentLength, initialFilamentArea, n],
          fluxVisualization[initialFilamentLength, initialFilamentArea, n, mobility]]
```

```
In[ ]:= electricPotentialAndFluxVisualization[
          initialFilamentLength, initialFilamentArea, n, mobility];
```



Creating a Dynamic, Iterative Simulation

Time Incrementation

```
In[ ]:= Δt = 0.0001;(*A very small time step is used to generate data
         points that very closely map onto a curve. Note that he value
         of Δt was changed from 0.0001 to 0.001 for the production of the
         Supplementary Videos to reduce file size and increase exporting speed.*)
```

Area Step

With each time increment, there will be a change in flux as well as a subsequent change in filament length. The changes in flux over time can be calculated by integrating the flux within a circle that encompasses the entire filament and its surrounding electrolyte at discrete times separated by a small time step of Δt.

```
In[ ]:= areaInWithinΔt[initialFilamentLength_, initialFilamentArea_, n_, mobility_, Δt_] :=
         areaInWithinΔt[initialFilamentLength, initialFilamentArea, n, mobility, Δt] =
          NIntegrate[flux[initialFilamentLength, initialFilamentArea, n, mobility][
              Cos[θ], Sin[θ]].{Cos[θ], Sin[θ]}, {θ, 0, Pi / 2}] Δt
```

The “area in” can be treated as equivalent to the flux times the nondimensionalized area of the atom of interest. Note that this disregards the effects of lattice packing and density on filament area. The

nondimensionalized area of the atom of interest can be approximated to equal the flux since all atomic radii, r_{atom}, are several orders of magnitude smaller than the nondimensionalized distance, d, that separates the two nanowires such that $1 \sim \frac{\pi r_{atom}^2}{d^2}$. This approximation is used for this model.

Length Step

The total filament length equals the initialFilamentLength + the change in filament length. The change in filament length equals the areaInWithinΔt, divided by the filament width (radius in this case). This is equivalent to the calculation of the length of a rectangle.

In[ ]:=
```
liveFilamentLength[initialFilamentLength_,
  initialFilamentArea_, n_, mobility_, Δt_] :=
 liveFilamentLength[initialFilamentLength, initialFilamentArea, n, mobility, Δt] =
  initialFilamentLength +
   (areaInWithinΔt[initialFilamentLength,
      initialFilamentArea, n, mobility, Δt] / (radius))
```

Dynamic Calculation of Flux and Filament Length

Iterative Filament Growth

```
In[•]:= filamentGrowthResultsTable =
          {{"time", "filament length", "filament area", "area in"},
           {0, initialFilamentLength, initialFilamentArea, 0}};
         (*Initial conditions that could be used as inputs if desired.*)
        Quiet@Module[{areaStep = 0, tempArea = initialFilamentArea,
            filamentLength = initialFilamentLength, time = 0, Δt = 0.0001},

          (*The filament grows until distance,
          d, between nanowires is reached or surpassed.*)
          While[filamentLength < 1,

           areaStep = areaInWithinΔt[filamentLength, tempArea, n, mobility, Δt];
           tempArea = tempArea + areaStep;
           filamentLength = liveFilamentLength[filamentLength, areaStep, n, mobility, Δt];
           time = time + Δt;

           (*The results are tabulated
            according to the headers of filamentGrowthResultsTable.*)
           AppendTo[filamentGrowthResultsTable,
            {time, filamentLength, tempArea, areaStep}]]]
```

This took about 10 minutes to run.

Example Output: Table

```
In[•]:= filamentGrowthResultsTable // TableForm;
```

"time"	"filament length"	"filament area"	"area in"
0	0.1	0.00892699	0
0.0001	0.102021	0.00902858	0.000101586
0.0002	0.104042	0.00913018	0.000101608
0.0003	0.106065	0.00923166	0.000101475
0.0004	0.108088	0.00933328	0.000101616
0.0005	0.110109	0.009435	0.000101724
0.0006	0.112134	0.00953678	0.00010178
0.0007	0.114158	0.00963861	0.000101827
0.0008	0.116184	0.00974047	0.000101868
0.0009	0.118208	0.00984237	0.000101898
0.001	0.120236	0.00994435	0.000101976
0.0011	0.122263	0.0100464	0.000102048
0.0012	0.124292	0.0101485	0.000102127

Printed by Wolfram Mathematica Student Edition

0.0013	0.126318	0.0102506	0.00010206
0.0014	0.12835	0.0103528	0.000102178
0.0015	0.130381	0.0104549	0.000102186
0.0016	0.132414	0.010557	0.000102101
0.0017	0.134447	0.0106594	0.000102354
0.0018	0.136481	0.0107618	0.000102374
0.0019	0.138515	0.0108642	0.000102405
0.002	0.140551	0.0109667	0.000102471
0.0021	0.142588	0.0110691	0.000102399
0.0022	0.144626	0.0111715	0.00010243
0.0023	0.146663	0.0112741	0.000102653
0.0024	0.148704	0.0113768	0.000102706
0.0025	0.150744	0.0114795	0.000102614
0.0026	0.152782	0.0115823	0.000102878
0.0027	0.154824	0.0116851	0.000102734
0.0028	0.156864	0.011788	0.000102973
0.0029	0.158908	0.011891	0.000102989
0.003	0.16095	0.0119939	0.000102897
0.0031	0.162996	0.0120971	0.000103181
0.0032	0.165045	0.0122002	0.000103137
0.0033	0.167092	0.0123035	0.000103228
0.0034	0.169138	0.0124067	0.000103209
0.0035	0.171184	0.01251	0.000103352
0.0036	0.173235	0.0126135	0.00010346
0.0037	0.175289	0.012717	0.000103502
0.0038	0.177343	0.0128205	0.00010355
0.0039	0.179399	0.0129242	0.000103608
0.004	0.181456	0.0130278	0.000103622
0.0041	0.183512	0.0131315	0.000103737
0.0042	0.185571	0.0132354	0.000103841
0.0043	0.18763	0.0133393	0.000103918
0.0044	0.189689	0.0134433	0.000103992
0.0045	0.191752	0.0135473	0.000104058
0.0046	0.193816	0.0136515	0.000104152
0.0047	0.195881	0.0137557	0.000104187
0.0048	0.197945	0.0138599	0.000104255
0.0049	0.200013	0.0139642	0.000104306
0.005	0.202082	0.0140687	0.000104484
0.0051	0.204153	0.0141732	0.000104527
0.0052	0.206225	0.0142779	0.000104659
0.0053	0.208298	0.0143826	0.000104701
0.0054	0.210373	0.0144872	0.000104623
0.0055	0.212448	0.0145921	0.000104858
0.0056	0.214526	0.014697	0.000104968
0.0057	0.216603	0.0148021	0.000105017

0.0058	0.21868	0.0149072	0.000105126
0.0059	0.220758	0.0150124	0.000105213
0.006	0.222842	0.0151177	0.000105304
0.0061	0.224927	0.0152231	0.000105408
0.0062	0.227015	0.0153286	0.000105445
0.0063	0.229105	0.0154339	0.000105361
0.0064	0.231196	0.0155396	0.000105641
0.0065	0.233285	0.0156453	0.000105747
0.0066	0.235378	0.0157511	0.00010579
0.0067	0.237471	0.0158568	0.000105727
0.0068	0.239567	0.0159628	0.000105971
0.0069	0.241664	0.0160689	0.000106102
0.007	0.243762	0.0161752	0.000106263
0.0071	0.245863	0.0162816	0.000106401
0.0072	0.247966	0.016388	0.000106397
0.0073	0.25007	0.0164946	0.000106607
0.0074	0.252177	0.0166012	0.000106589
0.0075	0.254285	0.0167079	0.000106713
0.0076	0.256395	0.0168147	0.000106825
0.0077	0.258507	0.0169215	0.00010686
0.0078	0.26062	0.0170286	0.000107013
0.0079	0.262736	0.0171358	0.000107227
0.008	0.264854	0.017243	0.000107227
0.0081	0.266973	0.0173503	0.000107318
0.0082	0.269094	0.0174578	0.000107466
0.0083	0.271218	0.0175653	0.000107546
0.0084	0.273343	0.017673	0.00010768
0.0085	0.275469	0.0177808	0.000107792
0.0086	0.277598	0.0178887	0.000107848
0.0087	0.279729	0.0179967	0.000108013
0.0088	0.281862	0.0181048	0.000108104
0.0089	0.283997	0.018213	0.000108196
0.009	0.286134	0.0183213	0.000108325
0.0091	0.288273	0.0184298	0.000108478
0.0092	0.290413	0.0185384	0.000108603
0.0093	0.292557	0.0186471	0.000108738
0.0094	0.294701	0.0187559	0.000108798
0.0095	0.296847	0.0188648	0.000108898
0.0096	0.298999	0.0189739	0.000109044
0.0097	0.301152	0.019083	0.000109166
0.0098	0.303303	0.0191923	0.000109309
0.0099	0.305459	0.0193017	0.000109365
0.01	0.307618	0.0194112	0.000109535
0.0101	0.309777	0.0195209	0.000109703
0.0102	0.311939	0.0196308	0.000109895
0.0103	0.314105	0.0197408	0.000109955

0.0104	0.316271	0.0198508	0.000110039
0.0105	0.31844	0.0199611	0.000110248
0.0106	0.320613	0.0200714	0.00011036
0.0107	0.322787	0.0201819	0.00011048
0.0108	0.324964	0.0202925	0.00011058
0.0109	0.327143	0.0204033	0.000110794
0.011	0.329326	0.0205142	0.000110907
0.0111	0.331509	0.0206252	0.000111051
0.0112	0.333696	0.0207364	0.000111129
0.0113	0.335884	0.0208477	0.000111295
0.0114	0.338076	0.0209592	0.000111537
0.0115	0.34027	0.0210708	0.000111566
0.0116	0.342467	0.0211826	0.000111801
0.0117	0.344666	0.0212944	0.000111817
0.0118	0.346867	0.0214064	0.000111966
0.0119	0.349072	0.0215185	0.000112144
0.012	0.351278	0.0216308	0.000112277
0.0121	0.353489	0.0217433	0.00011248
0.0122	0.355702	0.0218559	0.000112669
0.0123	0.357919	0.0219687	0.000112798
0.0124	0.360137	0.0220816	0.000112894
0.0125	0.362354	0.0221945	0.000112927
0.0126	0.364573	0.0223078	0.000113211
0.0127	0.3668	0.0224212	0.000113423
0.0128	0.369029	0.0225347	0.000113521
0.0129	0.371258	0.0226484	0.000113733
0.013	0.373493	0.0227624	0.00011392
0.0131	0.375731	0.0228765	0.000114117
0.0132	0.377972	0.0229905	0.00011398
0.0133	0.380218	0.0231046	0.000114153
0.0134	0.382465	0.0232192	0.000114556
0.0135	0.384715	0.023334	0.00011479
0.0136	0.386969	0.0234489	0.000114919
0.0137	0.389225	0.0235639	0.000115055
0.0138	0.391485	0.023679	0.00011506
0.0139	0.393747	0.0237944	0.000115424
0.014	0.396014	0.02391	0.000115599
0.0141	0.398283	0.0240259	0.000115889
0.0142	0.400556	0.0241418	0.000115872
0.0143	0.402832	0.0242577	0.000115977
0.0144	0.405113	0.024374	0.000116282
0.0145	0.407396	0.0244904	0.000116391
0.0146	0.409682	0.0246069	0.000116452
0.0147	0.411967	0.0247238	0.000116884
0.0148	0.414256	0.0248409	0.000117181

0.0149	0.416553	0.0249583	0.000117321
0.015	0.418853	0.0250757	0.000117476
0.0151	0.421151	0.0251936	0.000117816
0.0152	0.423455	0.0253113	0.000117747
0.0153	0.425765	0.0254294	0.000118126
0.0154	0.428071	0.0255478	0.00011835
0.0155	0.430389	0.0256663	0.000118555
0.0156	0.43271	0.025785	0.00011868
0.0157	0.435032	0.025904	0.000118998
0.0158	0.43736	0.026023	0.000118968
0.0159	0.439693	0.0261422	0.000119209
0.016	0.442026	0.0262618	0.0001196
0.0161	0.444366	0.0263814	0.000119604
0.0162	0.446711	0.0265014	0.000119982
0.0163	0.449059	0.0266216	0.000120209
0.0164	0.45141	0.026742	0.000120406
0.0165	0.453766	0.0268627	0.00012071
0.0166	0.456123	0.0269836	0.000120923
0.0167	0.458487	0.0271047	0.000121098
0.0168	0.460856	0.0272261	0.000121414
0.0169	0.463228	0.0273477	0.000121568
0.017	0.465605	0.0274695	0.000121841
0.0171	0.467985	0.0275916	0.000122055
0.0172	0.470367	0.0277136	0.000121989
0.0173	0.472755	0.0278359	0.000122357
0.0174	0.47515	0.0279587	0.000122742
0.0175	0.477549	0.0280817	0.000123015
0.0176	0.479955	0.028205	0.000123311
0.0177	0.482364	0.0283286	0.000123609
0.0178	0.484776	0.0284525	0.000123852
0.0179	0.487193	0.0285765	0.000124012
0.018	0.489615	0.0287009	0.000124397
0.0181	0.492042	0.0288254	0.000124534
0.0182	0.494474	0.0289502	0.00012479
0.0183	0.496908	0.0290752	0.000125047
0.0184	0.499351	0.0292005	0.000125264
0.0185	0.501799	0.0293261	0.000125585
0.0186	0.504249	0.029452	0.000125944
0.0187	0.506706	0.0295782	0.000126115
0.0188	0.509165	0.0297046	0.000126406
0.0189	0.51163	0.0298312	0.000126669
0.019	0.514101	0.0299582	0.000126964
0.0191	0.516577	0.0300855	0.000127306
0.0192	0.519058	0.0302132	0.000127674
0.0193	0.521543	0.0303411	0.000127898

0.0194	0.524034	0.0304692	0.000128147
0.0195	0.526533	0.0305974	0.000128203
0.0196	0.529033	0.0307262	0.000128799
0.0197	0.53154	0.0308553	0.000129083
0.0198	0.534054	0.0309848	0.000129539
0.0199	0.536573	0.0311146	0.000129761
0.02	0.539098	0.0312447	0.00013007
0.0201	0.541628	0.031375	0.000130338
0.0202	0.544164	0.0315057	0.000130736
0.0203	0.546706	0.0316368	0.000131051
0.0204	0.549254	0.0317681	0.000131343
0.0205	0.551807	0.0318999	0.000131744
0.0206	0.554367	0.0320318	0.000131925
0.0207	0.556933	0.0321642	0.000132358
0.0208	0.559504	0.0322969	0.000132747
0.0209	0.562082	0.0324299	0.000133023
0.021	0.564666	0.0325633	0.000133377
0.0211	0.567258	0.032697	0.00013368
0.0212	0.569853	0.032831	0.000134008
0.0213	0.572456	0.0329652	0.000134209
0.0214	0.575066	0.0331	0.000134828
0.0215	0.577682	0.0332352	0.000135161
0.0216	0.580305	0.0333707	0.00013553
0.0217	0.582935	0.0335067	0.00013592
0.0218	0.58557	0.033643	0.000136314
0.0219	0.588207	0.0337797	0.000136695
0.022	0.590852	0.0339167	0.000137063
0.0221	0.593505	0.034054	0.000137265
0.0222	0.596169	0.0341916	0.000137603
0.0223	0.598839	0.0343298	0.000138218
0.0224	0.601517	0.0344684	0.000138609
0.0225	0.604203	0.0346075	0.000139046
0.0226	0.606897	0.0347469	0.000139479
0.0227	0.609598	0.0348866	0.000139608
0.0228	0.612306	0.035027	0.000140423
0.0229	0.615023	0.0351674	0.000140463
0.023	0.617747	0.0353086	0.000141203
0.0231	0.620481	0.0354503	0.000141644
0.0232	0.623222	0.0355924	0.000142132
0.0233	0.62597	0.0357351	0.00014271
0.0234	0.628728	0.0358781	0.000142952
0.0235	0.631495	0.0360216	0.00014351
0.0236	0.634268	0.0361655	0.000143939
0.0237	0.637049	0.0363101	0.000144534
0.0238	0.639842	0.0364551	0.000145028

0.0239	0.642643	0.0366007	0.000145565
0.024	0.645452	0.0367466	0.000145938
0.0241	0.648269	0.0368931	0.000146495
0.0242	0.651095	0.0370401	0.000147024
0.0243	0.65393	0.0371876	0.000147531
0.0244	0.656774	0.0373356	0.000148002
0.0245	0.659627	0.0374841	0.000148476
0.0246	0.662492	0.0376332	0.000149046
0.0247	0.665365	0.0377827	0.000149543
0.0248	0.668242	0.0379329	0.000150234
0.0249	0.671137	0.0380838	0.000150811
0.025	0.674041	0.0382351	0.000151301
0.0251	0.676956	0.038387	0.00015191
0.0252	0.679882	0.0385395	0.000152502
0.0253	0.682817	0.0386925	0.000153077
0.0254	0.685761	0.0388463	0.000153746
0.0255	0.688717	0.0390006	0.000154318
0.0256	0.691684	0.0391557	0.000155054
0.0257	0.694663	0.0393112	0.00015557
0.0258	0.697652	0.0394674	0.000156213
0.0259	0.700652	0.039624	0.000156546
0.026	0.703665	0.0397816	0.000157604
0.0261	0.706684	0.0399399	0.00015826
0.0262	0.709721	0.0400988	0.000158957
0.0263	0.712771	0.0402581	0.000159296
0.0264	0.715833	0.0404186	0.000160538
0.0265	0.718901	0.0405796	0.000160932
0.0266	0.721989	0.0407415	0.000161941
0.0267	0.72509	0.0409043	0.00016279
0.0268	0.728204	0.0410673	0.000163032
0.0269	0.731333	0.0412315	0.000164169
0.027	0.734467	0.0413965	0.000164956
0.0271	0.737623	0.0415622	0.000165764
0.0272	0.740793	0.0417288	0.000166587
0.0273	0.74397	0.0418963	0.000167493
0.0274	0.747171	0.0420648	0.000168527
0.0275	0.750386	0.0422343	0.000169441
0.0276	0.753617	0.0424046	0.000170349
0.0277	0.756868	0.0425757	0.000171068
0.0278	0.760131	0.0427476	0.000171931
0.0279	0.763411	0.0429206	0.000172929
0.028	0.766707	0.0430945	0.000173941
0.0281	0.770021	0.0432699	0.00017539
0.0282	0.773353	0.0434461	0.000176213
0.0283	0.776702	0.0436232	0.000177077
0.0284	0.78007	0.0438015	0.000178303

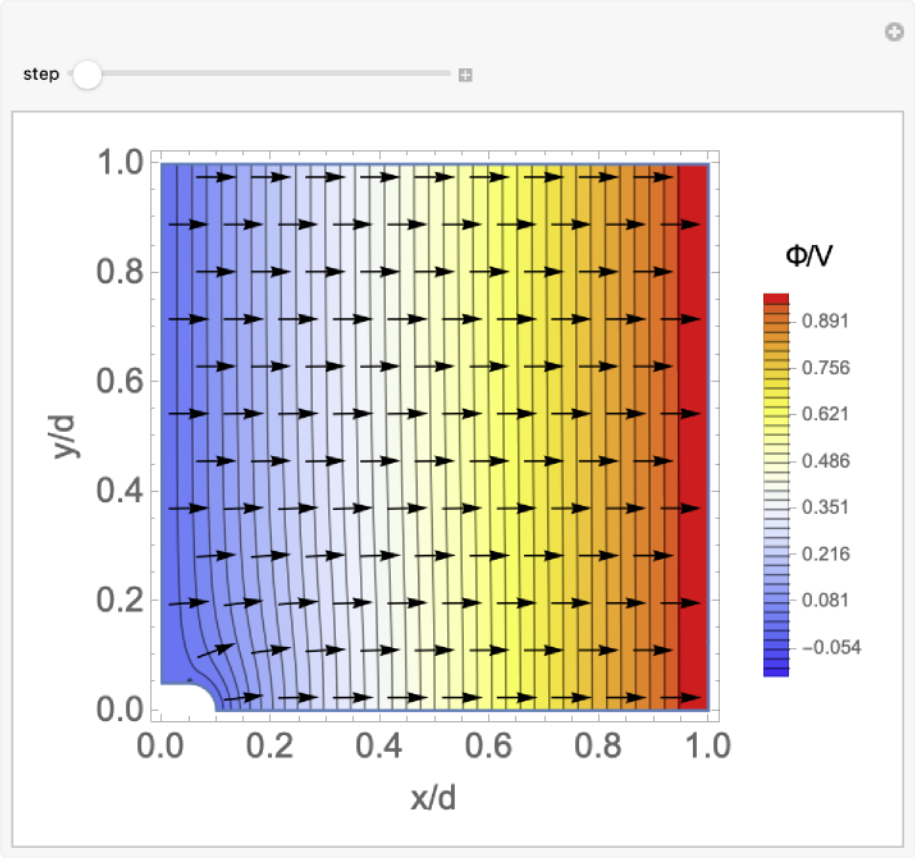
0.0285	0.783455	0.0439807	0.000179213
0.0286	0.78686	0.0441613	0.000180619
0.0287	0.790284	0.0443423	0.000181007
0.0288	0.793727	0.044525	0.000182641
0.0289	0.797191	0.0447084	0.000183407
0.029	0.800676	0.0448935	0.000185127
0.0291	0.804182	0.0450803	0.000186766
0.0292	0.807711	0.0452674	0.000187136
0.0293	0.811266	0.0454565	0.000189069
0.0294	0.814842	0.0456469	0.000190438
0.0295	0.818439	0.045839	0.000192077
0.0296	0.822061	0.0460324	0.00019346
0.0297	0.825709	0.0462274	0.000194978
0.0298	0.829386	0.0464238	0.00019641
0.0299	0.83309	0.0466219	0.000198096
0.03	0.836821	0.0468218	0.000199862
0.0301	0.840578	0.0470231	0.0002013
0.0302	0.844371	0.0472255	0.000202378
0.0303	0.848188	0.0474298	0.000204372
0.0304	0.852023	0.047636	0.000206146
0.0305	0.855905	0.0478451	0.000209088
0.0306	0.859819	0.048056	0.000210899
0.0307	0.863768	0.0482691	0.000213121
0.0308	0.867753	0.0484848	0.000215719
0.0309	0.87176	0.0487023	0.000217448
0.031	0.875824	0.048922	0.000219777
0.0311	0.879909	0.0491448	0.000222797
0.0312	0.884047	0.0493705	0.000225631
0.0313	0.888228	0.0495981	0.00022764
0.0314	0.892455	0.049829	0.000230933
0.0315	0.896725	0.0500631	0.000234064
0.0316	0.901047	0.0503005	0.000237356
0.0317	0.905424	0.0505414	0.000240982
0.0318	0.909856	0.0507862	0.000244791
0.0319	0.914351	0.0510349	0.000248705
0.032	0.918912	0.0512881	0.00025318
0.0321	0.923541	0.0515461	0.000257972
0.0322	0.928241	0.0518092	0.000263103
0.0323	0.933028	0.0520783	0.000269079
0.0324	0.937899	0.0523535	0.000275239
0.0325	0.942865	0.0526357	0.000282182
0.0326	0.947939	0.0529259	0.000290225
0.0327	0.953135	0.0532254	0.000299508
0.0328	0.958461	0.0535352	0.000309763
0.0329	0.963951	0.0538579	0.000322765

0.033	0.969621	0.0541957	0.000337753
0.0331	0.975512	0.0545537	0.000358036
0.0332	0.981685	0.0549383	0.00038459
0.0333	0.98824	0.0553623	0.000423959
0.0334	0.995363	0.0558556	0.000493297
0.0335	1.00383	0.0565537	0.000698148

Dynamic Simulation

Interactive Simulation

In[ ]:=
```
Manipulate[Show[electricPotentialVisualization[
    filamentGrowthResultsTable[[step]][[2]], filamentGrowthResultsTable[[step]][[3]], n],
  fluxVisualization[filamentGrowthResultsTable[[step]][[2]],
   filamentGrowthResultsTable[[step]][[3]], n, mobility], BaseStyle → 18],
 {step, 2, Length[filamentGrowthResultsTable], 1}]
```



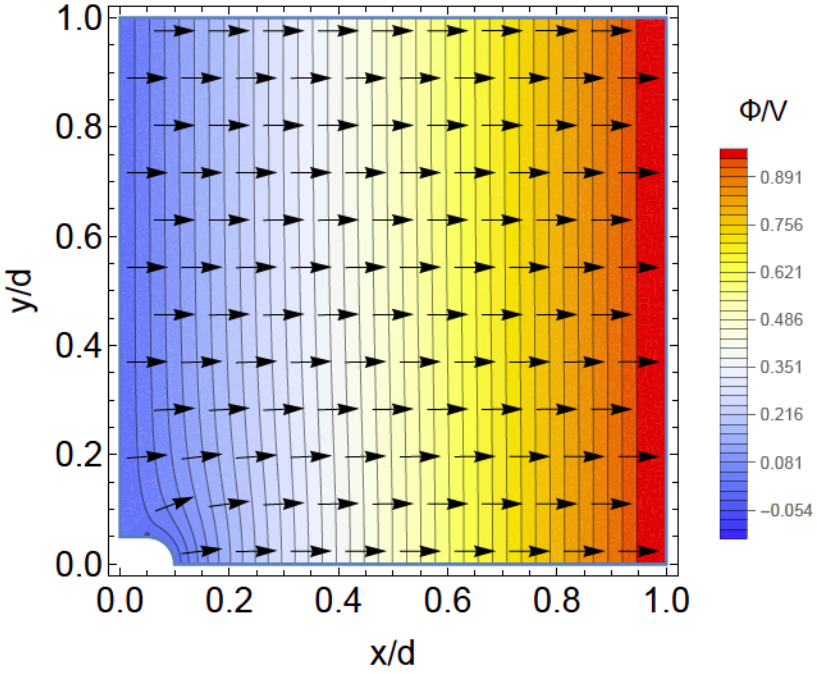
Tabulated Snapshots

```
In[•]:= filamentGrowthResultsSnapshotTable =
  Table[Show[electricPotentialVisualization[filamentGrowthResultsTable[[step]][[2]],
     filamentGrowthResultsTable[[step]][[3]], n],
    fluxVisualization[filamentGrowthResultsTable[[step]][[2]],
     filamentGrowthResultsTable[[step]][[3]], n, mobility], BaseStyle → 18],
   {step, 2, Length[filamentGrowthResultsTable], 1}];
```

Output Example: Visualization

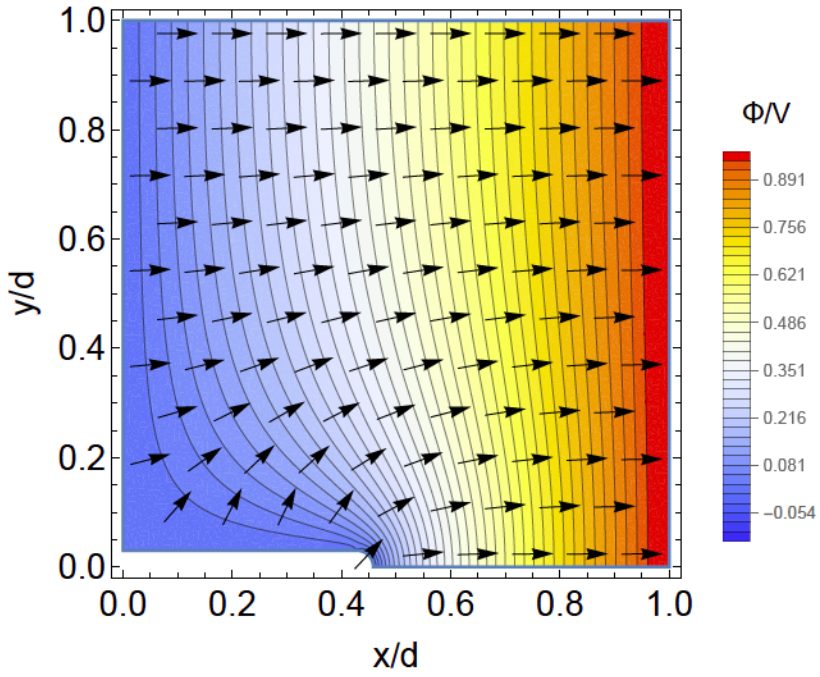
At $\tau^{non} = 0$:

```
In[•]:= snapshot1 = filamentGrowthResultsSnapshotTable[[1]];
```



At $\tau^{non} = \tau^{non,tot/2}$:

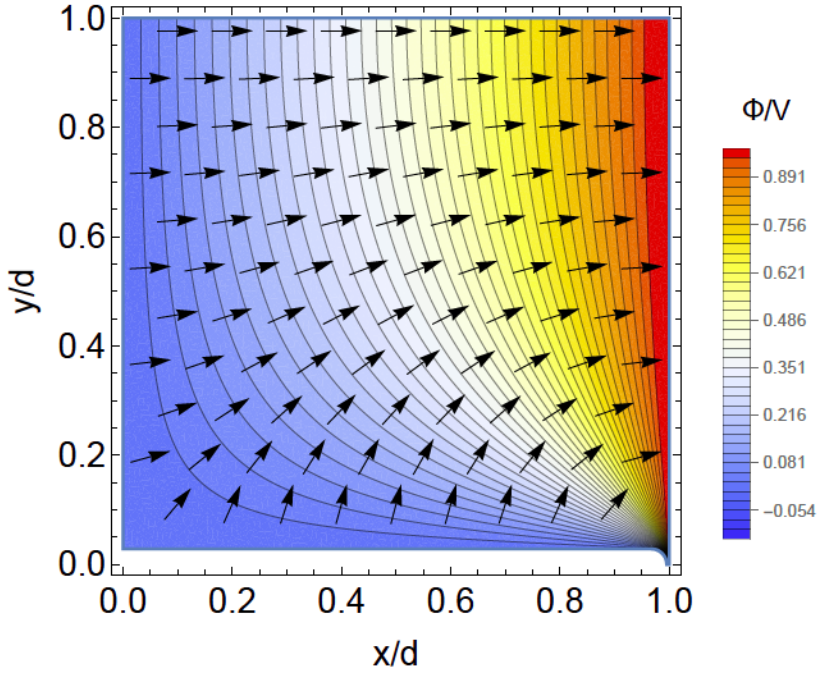
```
In[•]:= snapshot2 = filamentGrowthResultsSnapshotTable[
    Round[Length[filamentGrowthResultsSnapshotTable] / 2]];
```



If the total number of coordinates is odd, then the coordinate directly following what would be $\tau^{non} = \tau^{non,tot/2}$ is chosen. This minute overestimation of filament growth can be considered negligible due to the small time step $\Delta t = 0.0001$ used here.

At $\tau^{non} = \tau^{non,tot}$:

In[•]:=
```
snapshot3 = filamentGrowthResultsSnapshotTable[[-2]]
```



The coordinate chosen for showcasing $\tau^{non} = \tau^{non,tot}$ is one time step past spanning the distance, d, separating the two nanowires. This miniscule separation prevents the calculations from undershooting

the distance, d, since the simulation does not capture when exactly the system reaches $\tau^{non,tot}$ but rather finds the time at which the total nondimensionalized filament length exceeds 1. Given the small time step Δt = 0.0001, this effect should be negligible.

Exporting

Videos

Export using a desired file path in quotation marks, taking special care to include backslashes in proper positions. Both the "desiredFileName" and file type ".mp4" can be changed as desired. An example file path is shown below.

```
Export["/Users/saravfernandez/Desktop/UGThesis/desiredFileName.mp4",
  filamentGrowthResultsSnapshotTable[[1 ;; -1]];
```

Using -1 includes the coordinate that overshoots d, so changing it from -1 to -2 can be done if desired.

Images

As before, export using a desired file path in quotation marks, taking special care to include back-slashes in proper positions. Both the "desiredFileName" and file type ".tiff" can be changed as desired. An example file path is shown below with the example image "snapshot1" getting exported. This code can be adapted to export any images.

```
Export["/Users/saravfernandez/Desktop/UGThesis/desiredFileName.tiff", snapshot1];
```

Filament Length as a Function of Time

NonLinear Model

Finding Nonlinear Fit

Here is a reminder of the table headings for filamentGrowthResultsTable.

In[•]:=
```
filamentGrowthResultsTable[[1]];
```

{time,filament length,filament area,area in}

Here, "nlm" indicates that this is a nonlinear model.

```
In[•]:= nlmLengthVsTime[table_] := nlmLengthVsTime[table] =
          NonlinearModelFit[table[[2 ;; -1(*to avoid undershooting d*),
              1(*time*) ;; 2(*filament length*)]],

           (*This is the expression that is used to find a nonlinear fit. It is possible
            that another expression would be better suited for describing this system,
           but this was the closest match investigated in this work.*)
           a Exp[b * Sqrt[x]] + c, {a, b, c}, x, MaxIterations → Infinity]

        lengthVsTimePlot[table_] := lengthVsTimePlot[table] =
          Show[ListPlot[table[[1 ;; -1(*to avoid undershooting d*),
              1(*time*) ;; 2(*filament length*)]],

            Frame → True, FrameLabel → {"τμ₀N_A dV_app", "x/d"},
            PlotLabel → "Filament Length vs. Time",
            PlotStyle → {PointSize[0.02], Opacity[0.1], Blue}, LabelStyle →
             Directive[FontFamily → "Helvetica", Black, FontSize → 18], PlotRange → Full],

           lengthVsTime[table] =
            Plot[nlmLengthVsTime[table][x], {x, 0, 1}, PlotRange → {{0, 1}, {0, 1}},
             PlotStyle → Blue], ImageSize → Large, BaseStyle → 18]
```

Here, the tabulated items in filamentGrowthResultsTable can be saved to a new variable so as to avoid re-running the simulation for the particular initial conditions used. In the interest of brevity, the presentation of this work will exclude such saved values.

Example Output: Equation and Plot

Here is the fitted equation.

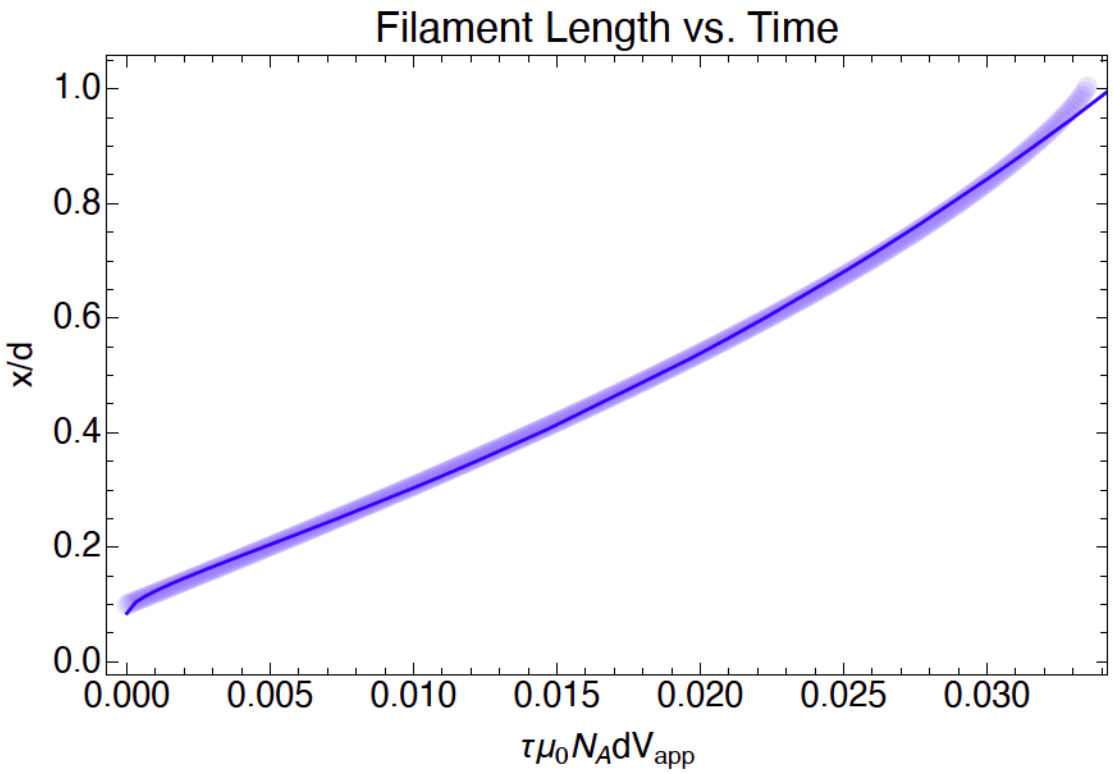
```
In[•]:= nonDimensionalizedLengthVsTime[table_] :=
         nonDimensionalizedLengthVsTime[table] = Normal[nlmLengthVsTime[table]]

In[•]:= nonDimensionalizedLengthVsTime[filamentGrowthResultsTable];
```

$$0.0169031 + 0.0676949\, e^{14.4392\,\sqrt{x}}$$

Here is the plot of the fitted equation.

```
In[•]:= lengthVsTimePlot[filamentGrowthResultsTable];
```

In[ ]:=



Statistical Validation

Any number of statistical tests can be run using Mathematica's extensive functionality. Here, two examples are provided, namely AdjustedRSquared and ParameterConfidenceIntervals.

Here is the adjustedRSquared.

In[ ]:=
```
nlmLengthVsTime[filamentGrowthResultsTable]["AdjustedRSquared"];
```

0.999864

Here are the ParameterConfidenceIntervals.

In[ ]:=
```
nlmLengthVsTime[filamentGrowthResultsTable]["ParameterConfidenceIntervals"];
```

{{0.0655731,0.0698166},{14.2843,14.594},{0.0124734,0.0213327}}

Computational Validation Using Realistic System Parameters

As an example, a system consisting of two perpendicular silver (Ag) nanowires in an SiO_2 electrolyte with ion mobility $\mu_0 = 3.178 \cdot 10^{-11} \frac{m^2}{Vs}$ separated by a distance d = 50 nm that undergoes an applied voltage V_{app}=1.2 V is considered. μ will be used instead of μ_0 going forward for typing ease and to avoid issues with the zero.

Dimensionalize the Equation

Start with the nonDimensionalizedLengthVsTime equation and treat x as τ^{non}. Replace τ^{non} with its equivalent physical parameters based on $\tau^{non}=\tau N_A\, \mu d V_{app}$.

```
In[ ]:= dimensionalizedLengthVsTime[table_] := dimensionalizedLengthVsTime[table] =
          nonDimensionalizedLengthVsTime[table] * d /. {x → τ * avogadroN * μ * d * V_app}
```

```
In[ ]:= dimensionalizedLengthVsTime[filamentGrowthResultsTable];
```

$$d\left(0.0169031 + 0.0676949\, e^{14.4392\sqrt{\text{avogadroN}\, d\, \mu\, \tau\, V_{app}}}\right)$$

Then replace τ^{non} with its equivalent physical parameters and convert to τ, time with units of s, with values using the relevant equation.

```
In[ ]:= numericalAvogadroN = 6.02214076 * 10^23; (*Avogadro's Number*)
```

```
In[ ]:= numericalLengthVsTime[table_] :=
         numericalLengthVsTime[table] = dimensionalizedLengthVsTime[table] /.
           {μ → 3.178 * 10^(-11), d → 50 * 10^(-9), V_app → 1.2, avogadroN → numericalAvogadroN}
```

```
In[ ]:= numericalLengthVsTime[filamentGrowthResultsTable];
```

$$\frac{0.0169031+0.0676949\, e^{15\,472.8\sqrt{\tau}}}{20\,000\,000}$$

Find $\tau^{non,max}$

```
In[ ]:= τNonMax[table_] := τNonMax[table] = table[[-1(*to avoid undershooting d*), 1(*time*)]]
```

```
In[ ]:= τNonMax[filamentGrowthResultsTable];
```

0.0335

Convert to Find τ^{max} in units of s

```
In[ ]:= τMax[table_] :=
         τMax[table] = τNonMax[table] * (1 / (μ * avogadroN * d * V_app)) (*equivalent to τ*) /.
           {μ → 3.178 * 10^(-11), d → 50 * 10^(-9), V_app → 1.2, avogadroN → numericalAvogadroN}
```

```
In[ ]:= τMax[filamentGrowthResultsTable];(*in units of s*)
```

2.91735×10^{-8}

Find xMax

```
In[ ]:= xMax[table_] :=
         xMax[table] =
          table[[-1(*to exclude coordinate that undershoots d*)]][[2(*filament length*)]] *
            d /. {d → 50 * 10^(-9)}
```

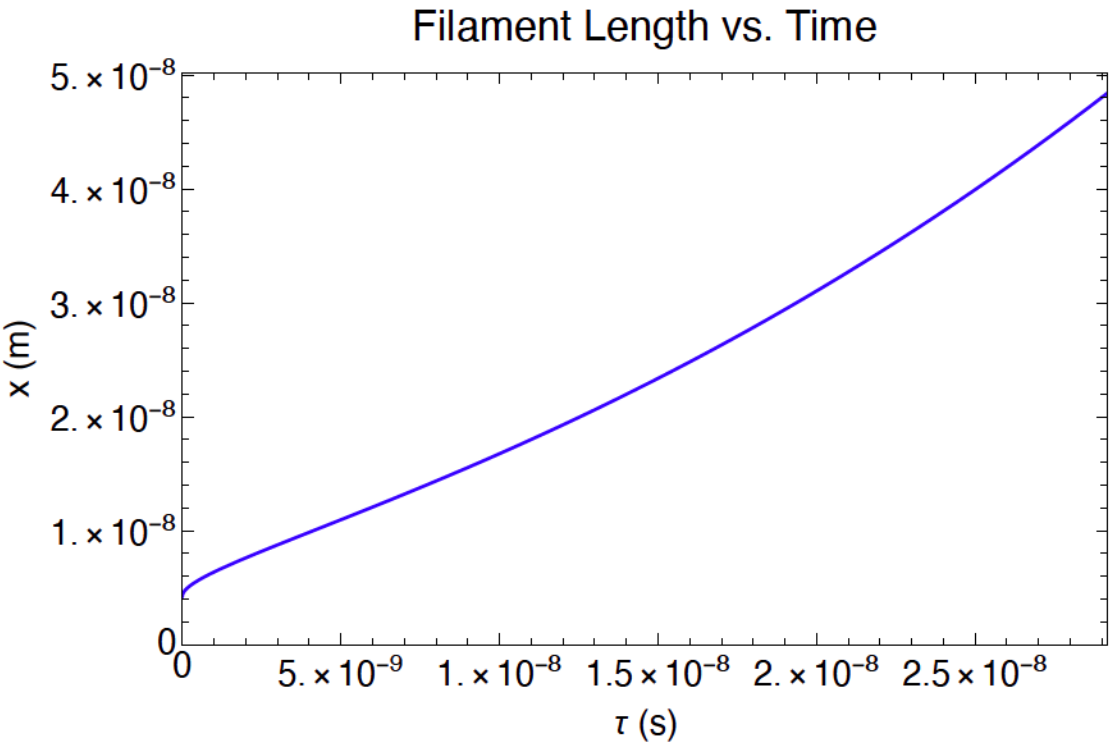
In[ ]:=
```
xMax[filamentGrowthResultsTable];(*in units of m*)
```

5.01914×10^{-8}

Plot Numerically

In[ ]:=
```
numericalLengthVsTimePlot[table_] := numericalLengthVsTimePlot[table] =
  Plot[numericalLengthVsTime[table], {τ, 0, τMax[table]},
   PlotRange → {{0, τMax[table]}, {0, xMax[table](*total length*)}},
   PlotStyle → Blue, Frame → True, FrameLabel → {"τ (s)", "x (m)"},
   PlotLabel → "Filament Length vs. Time",
   LabelStyle → Directive[FontFamily → "Helvetica", Black, FontSize → 18],
   ImageSize → Large](*xmax should equal d*)
```

In[ ]:=
```
numericalLengthVsTimePlot[filamentGrowthResultsTable];
```



Relationships Between Variables

Initial Conditions

Here is a reminder of τNonMax.

In[ ]:=
```
τNonMax[filamentGrowthResultsTable];
```

0.0335

```
In[ ]:= toτConversionFactor[μFactor_, dFactor_, vFactor_] :=
         toτConversionFactor[μFactor, dFactor, vFactor] = (1 / (avogadroN * μ * d * v)) /.
           {avogadroN → numericalAvogadroN, μ → μFactor * 3.178 * 10^(-11),
            d → dFactor * 50 * 10^(-9), v → vFactor * 1.2} (*Multiply this by τ^non to get τ.*)

In[ ]:= toτConversionFactor[1, 1, 1];
```

8.70851×10^{-7}

```
In[ ]:= calculatedTau[table_, μFactor_, dFactor_, vFactor_] :=
         calculatedTau[table, μFactor, dFactor, vFactor] =
          τNonMax[table] * toτConversionFactor[μFactor, dFactor, vFactor]

In[ ]:= calculatedTau[filamentGrowthResultsTable, 1, 1, 1];
```

2.91735×10^{-8}

Doubling Applied Voltage

```
In[ ]:= calculatedTau[filamentGrowthResultsTable, 1, 1, 2];
```

1.45868×10^{-8}

Doubling applied voltage cuts τ in half, meaning the filament grows faster and takes less time with higher voltage. This is also the case for mobility and distance it seems as they output the same quantity.

Doubling Mobility

```
In[ ]:= calculatedTau[filamentGrowthResultsTable, 2, 1, 1];
```

Doubling Distance

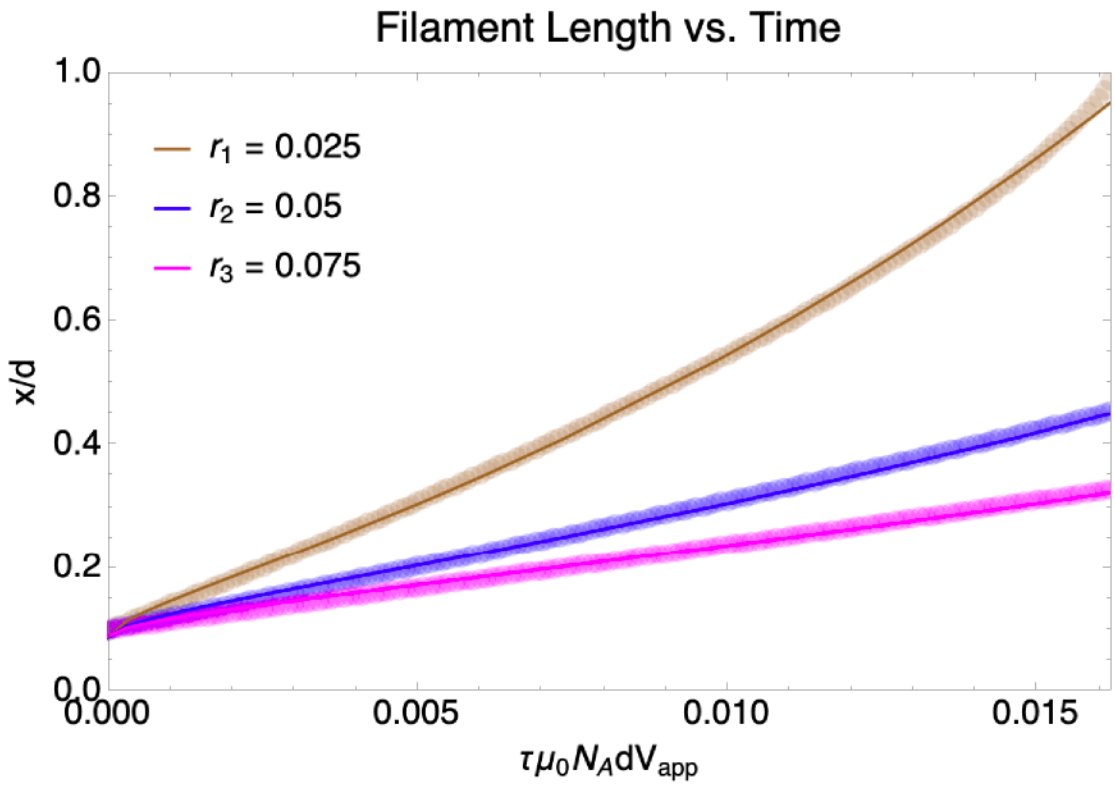
```
In[ ]:= calculatedTau[filamentGrowthResultsTable, 1, 2, 1];
```

Effect of Changing Initial Filament Conditions

```
In[ ]:= radiusComparisonFilamentVsLengthPlot[table1_, table2_, table3_] :=
         radiusComparisonFilamentVsLengthPlot[table1, table2, table3] =
          Plot[{nonDimensionalizedLengthVsTime[table1], nonDimensionalizedLengthVsTime[
              table2], nonDimensionalizedLengthVsTime[table3]},
            {x, 0, 1}, PlotStyle → {Brown, Blue, Magenta}, Frame → True,
            FrameLabel → {"τμ₀N_A dV_app", "x/d"}, PlotRange → {{0, τNonMax[table1]
               (*The longest table will be the one with the smallest radius.*)}, {0, 1}},
            PlotStyle → {Brown, Blue, Magenta}, ImageSize → Large,
            PlotLabel → "Filament Length vs. Time", LabelStyle →
             Directive[FontFamily → "Helvetica", Black, FontSize → 18, BaseStyle → 18],
            PlotLegends → Placed[{"r₁ = 0.025", "r₂ = 0.05", "r₃ = 0.075"}, {0.15, .78}],
            ImageSize → Large]
```

To make this work, tabulated values need to be generated for multiple radii. The ones used here are noted in the plot. In the interest of brevity, the presentation of this work will exclude tabulated values beyond the first case, meaning table1 and table3 are not associated with generated tables.

In[•]:=
```
radiusComparisonFilamentVsLengthPlot[table1, filamentGrowthResultsTable, table3];
```



Current as a Function of Time

Nonlinear Model

Finding Nonlinear Fit

Here is a reminder of the table headings for filamentGrowthResultsTable.

In[•]:=
```
filamentGrowthResultsTable[[1]];
```

{time,filament length,filament area,area in}

Here, “nlm” indicates that this is a nonlinear model.

```
In[ ]:= nlmCurrentVsTime[table_] :=
 nlmCurrentVsTime[table] = NonlinearModelFit[table[[3(*to avoid unrealistic jump
      of current from zero to first nonzero value contained in position 2*) ;; -1
     (*to avoid undershooting d*), {1(*time*), 4(*area in*)}]],

   (*This is the expression that is used to find a nonlinear fit. It is possible
    that another expression would be better suited for describing this system,
   but this was the closest match investigated in this work.*)
   a * Exp[b * x^c] + d + e * x, {a, b, c, d, e}, x, MaxIterations → Infinity]

currentVsTimePlot[table_] :=
 currentVsTimePlot[table] = Show[ListPlot[table[[3(*to avoid unrealistic jump of
       current from zero to first nonzero value contained in position 2*) ;; -1
      (*to avoid undershooting d*), {1(*time*), 4(*area in*)}]],

    Frame → True, FrameLabel → {"τμ₀N_AdV_app", "I / (μ₀ N_A dV_app)"},

    PlotLabel → "Ionic Current vs. Time",
    PlotStyle → {PointSize[0.02], Opacity[0.1], Blue}, LabelStyle →
     Directive[FontFamily → "Helvetica", Black, FontSize → 18], PlotRange → Full],

   currentVsTime[table] =
    Plot[nlmCurrentVsTime[table][x], {x, 0, 1}, PlotRange → {{0, 1}, {0, 1}},
     PlotStyle → Blue], ImageSize → Large, BaseStyle → 18]
```

Example Output: Equation and Plot

Here is the fitted equation.

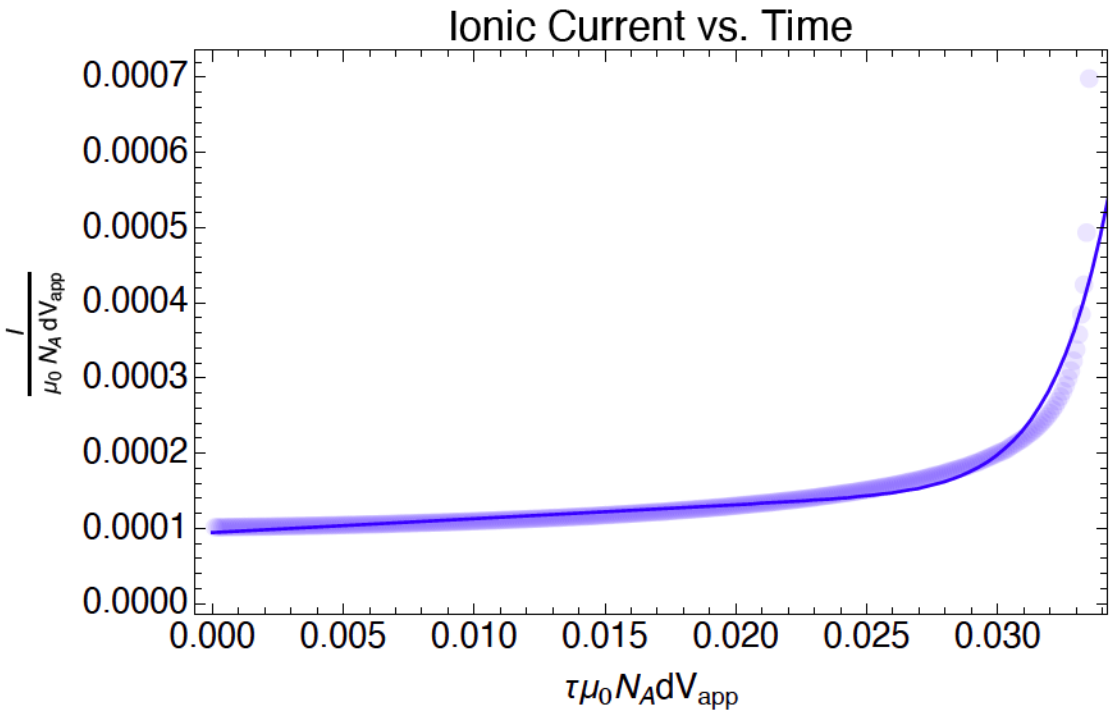
```
In[ ]:= nonDimensionalizedCurrentVsTime[table_] :=
 nonDimensionalizedCurrentVsTime[table] = Normal[nlmCurrentVsTime[table]]
```

```
In[ ]:= nonDimensionalizedCurrentVsTime[filamentGrowthResultsTable];
```

$$2.47955 \times 10^9 - 2.47955 \times 10^9\, e^{-1.51582 \times 10^{10}\, x^{15.6884}} + 0.00184416\, x$$

Here is the plot of the fitted equation.

```
In[ ]:= currentVsTimePlot[filamentGrowthResultsTable];
```



Statistical Validation

Any number of statistical tests can be run using Mathematica's extensive functionality. Here, two examples are provided, namely AdjustedRSquared and ParameterConfidenceIntervals.

Here is the adjustedRSquared.

In[•]:=
```
nlmCurrentVsTime[filamentGrowthResultsTable]["AdjustedRSquared"];
```

0.98618

Here are the ParameterConfidenceIntervals.

In[•]:=
```
nlmCurrentVsTime[filamentGrowthResultsTable]["ParameterConfidenceIntervals"];
```

$\{\{-2.47955\times10^{9}, -2.47955\times10^{9}\},$
$\{-1.51582\times10^{10}, -1.51582\times10^{10}\}, \{15.6734, 15.7033\},$
$\{2.47955\times10^{9}, 2.47955\times10^{9}\}, \{0.00160001, 0.00208832\}\}$

Computational Validation Using Realistic System Parameters

As an example, a system consisting of two perpendicular silver (Ag) nanowires in an SiO_2 electrolyte with ion mobility μ_0= 3.178·10^{-11} $\frac{m^2}{Vs}$ separated by a distance d = 50 nm that undergoes an applied voltage V_{app}=1.2 V is considered. μ will be used instead of $\mu_0$going forward for typing ease and to avoid issues with the zero.

Printed by Wolfram Mathematica Student Edition

Dimensionalize the Equation

Start with the nonDimensionalizedLengthVsTime equation and treat x as τ^{non}. Replace τ^{non} with its equivalent physical parameters based on $I^{non}=I\tau$.

```
In[ ]:= dimensionalizedCurrentVsTime[table_] :=
 nonDimensionalizedCurrentVsTime[table] / (1 / (avogadroN * μ * d * V_app)) /.
  {x → τ * avogadroN * μ * d * V_app}
```

```
In[ ]:= dimensionalizedCurrentVsTime[filamentGrowthResultsTable];
```

$$\text{avogadroN d } \mu \text{ V}_{\text{app}} \left(2.47955\times 10^9 - 2.47955\times 10^9\, e^{-1.51582\times 10^{10}\,(\text{avogadroN d } \mu\ \tau\ \text{V}_{\text{app}})^{15.6884}} + 0.00184416\ \text{avogadroN d } \mu\ \tau\ \text{V}_{\text{app}}\right)$$

Then replace τ^{non} with its equivalent physical parameters and convert to τ, time with units of s, with values using the relevant equation.

```
In[ ]:= numericalAvogadroN = 6.02214076 * 10^23; (*Avogadro's Number*)
```

```
In[ ]:= numericalCurrentVsTime[table_] :=
 numericalCurrentVsTime[table] = dimensionalizedCurrentVsTime[table] /.
   {μ → 3.178 * 10^(-11), d → 50 * 10^(-9), V_app → 1.2, avogadroN → numericalAvogadroN}
```

```
In[ ]:= numericalCurrentVsTime[filamentGrowthResultsTable];
```

$$1.1483\times 10^6 \left(2.47955\times 10^9 - 2.47955\times 10^9\, e^{-1.79106\times 10^{105}\,\tau^{15.6884}} + 2117.66\ \tau\right)$$

Find $I^{non,max}$

```
In[ ]:= iNonMax[table_] :=
 iNonMax[table] = table[[-1(*to avoid undershooting d*), 4(*area in*)]]
```

```
In[ ]:= iNonMax[filamentGrowthResultsTable];
```

```
Out[ ]= 0.000698148
```

Convert to Find τ^{max} in units of s

```
iMax[table_] := iMax[table] = iNonMax[table] / (1 / (μ * avogadroN * d * V_app) (*τ*))
   (*equivalent to dividing by τ*) /.
   {μ → 3.178 * 10^(-11), d → 50 * 10^(-9), V_app → 1.2, avogadroN → numericalAvogadroN}
```

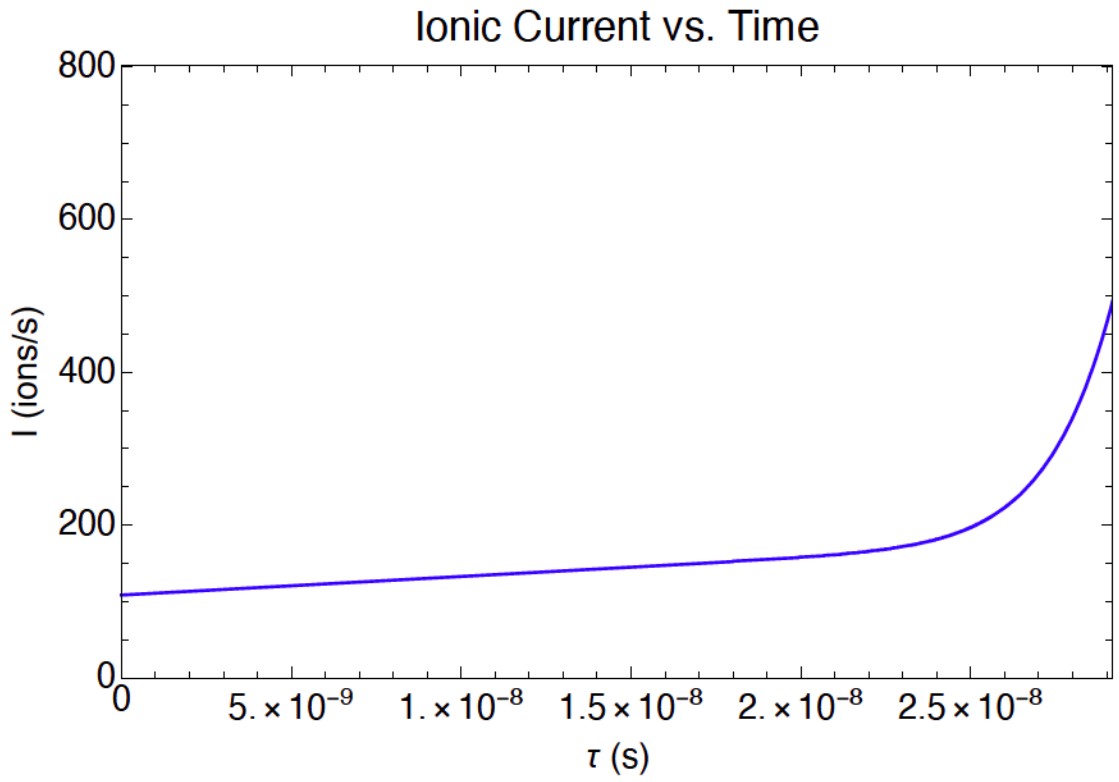
```
In[ ]:= iMax[filamentGrowthResultsTable];(*in units of ions/s*)
```

```
Out[ ]= 801.685
```

Plot Numerically

```
In[ ]:= numericalCurrentVsTimePlot[table_] := numericalCurrentVsTimePlot[table] =
          Plot[numericalCurrentVsTime[table], {τ, 0, τMax[table]},
           PlotRange → {{0, τMax[table]}, {0, iMax[table](*total length*)}},
           PlotStyle → Blue, Frame → True, FrameLabel → {"τ (s)", "I (ions/s)"},
           PlotLabel → "Ionic Current vs. Time", LabelStyle →
            Directive[FontFamily → "Helvetica", Black, FontSize → 18], ImageSize → Large]
```

```
In[ ]:= numericalCurrentVsTimePlot[filamentGrowthResultsTable];
```



Effect of Different Variables on Max Current

Initial Conditions

Here is a reminder of iNonMax.

```
In[ ]:= iNonMax[filamentGrowthResultsTable];
```

0.000698148

```
In[ ]:= toIConversionFactor[μFactor_, dFactor_, vFactor_] :=
         toIConversionFactor[μFactor, dFactor, vFactor] = (avogadroN * μ * d * v) /.
           {avogadroN → numericalAvogadroN, μ → μFactor * 3.178 * 10^(-11),
            d → dFactor * 50 * 10^(-9), v → vFactor * 1.2} (*Multiply this by I^non to get I.*)
```

```
In[•]:= toIConversionFactor[1, 1, 1];
```

1.1483×10^{6}

```
In[•]:= ionicIMax[table_, μFactor_, dFactor_, vFactor_] :=
         ionicIMax[table, μFactor, dFactor, vFactor] =
          iNonMax[table] * toIConversionFactor[μFactor, dFactor, vFactor]
```

```
In[•]:= ionicIMax[filamentGrowthResultsTable, 1, 1, 1];(*in units of ions/s*)
```

801.685

Doubling Applied Voltage

```
In[•]:= ionicIMax[filamentGrowthResultsTable, 1, 1, 2];
```

1603.37

Doubling applied voltage doubles I. This is also the case for mobility and distance it seems as they output the same quantity.

Doubling Mobility

```
In[•]:= ionicIMax[filamentGrowthResultsTable, 2, 1, 1];
```

Doubling Distance

```
In[•]:= ionicIMax[filamentGrowthResultsTable, 1, 2, 1];
```

Current in Amperes (A)

```
In[•]:= cationCharge = 1.602 * 10^(-19);(*in units of C*)
```

```
In[•]:= electricIMax[table_, μFactor_, dFactor_, vFactor_] :=
         electricIMax[table, μFactor, dFactor, vFactor] =
          ionicIMax[table, μFactor, dFactor, vFactor] * cationCharge
```

```
In[•]:= electricIMax[filamentGrowthResultsTable, 1, 1, 1];(*in units of Amperes (A)*)
```

1.2843×10^{-16}

Printed by Wolfram Mathematica Student Edition