**Modeling and simulation final report**

**Nasrullah Rami Hamad**

**Dr. Murad Yaghi**

Section 1

**Task 1:**

Question 1:

**Single neurons and its function and principles**

Neuroscientists use single-neurons and their models to identify the information understanding process of such neurons. The models can be used to comprehend such behavior by exploring the brain and its electrical behavior, using mathematical equations and other functions. Some of the principles of these neurons:

* The single-neuron models can be used to balance the complexity of the brain neural networks, and the way it is done depends on the model used, McCulloch-Pitt’s neuron uses a binary threshold to abstract the complexity. While some complex models like Hodgkin-Huxley model uses ion channels to gain a more realistic action of electrical activity
* The neurons are able to receive inputs from the rest of the neurons, these inputs can either increase the chance of a neuron firing (spike) or decrease it. The way these models can comprehend this integration is by taking the sum of the input weights.
* As mentioned before, a spike can be generated by increasing the chance of the neuron firing, the neuron firing can be in the shape of an electrical signal that travels inside the neuron space. Different models treat the thresholding uniquely.
* There are plenty of factors that can contribute a bit of randomness when neurons start firing the spikes, some of these factors are the noise of channels, and what mechanisms are used in the model.

To study such behaviors and principles neuroscientists and researchers use several tools and ways to investigate the models. The tools can either be computational software and tools, or simulations. Mathematics and analytical techniques can also be used to study the single-neuron models.

Question 2:

**benefits of applying modeling and simulation for simulating the neuronal dynamics:**

Modeling and simulation can enhance the studying of the neuronal dynamics and features for scientists and researchers. This way can generate a whole bunch of benefits and a deeper dive into the world of neuronal dynamics. Some of these benefits are:

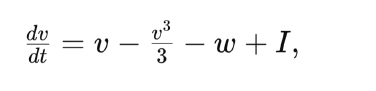
* Enhances the understanding and investigations of neural behaviors, since using simulations can guarantee using real neurons as a form of our understanding, we can manipulate the system and its variables to study further into the model, and its dynamics. It gives neuroscientists a better understanding of these mechanisms by looking at different perspectives in the simulation.
* Reduces the cost and time of waiting for scenarios to happen, since simulation can study different approaches of the modeling procedure, the time can be constantly lower than waiting for the real neurons to do anything new in particular.
* Modeling and simulation can also help by bringing new and significant technologies of combined networks of neurons that wouldn’t be possible by studying single-neuron models alone. This can help create models that can simulate the human brain and understand properties like decision-making or memorizing for humans.
* Usage of such computational software and tools for new investigations of how neurons work together can help guide scientists into getting a further understanding of how neural networks inside the brain work, so they can focus on creating new drugs and medicine.

Modeling and simulation are very vast yet an interesting field to integrate into the neuronal dynamic world. It can generate smarter and more efficient readings of the neural models. Which can be difficult and time-consuming for humans normally. Such simulations can help investigate and discover millions of scenarios in the human brain with less time than usual.

Question 3:

**Simulate each equation using Python and plot the results for each equation.**

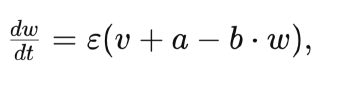
***The equations have been implemented in a python code***



The first equation describes the rate of change of v with respect to time (t). The factors that influence this rate of change are: v, v^3/3, w, and I. Each variable will influence the rate of change based on its decrease or increase inside the equation. Here are the explanation of all the terms inside this equation:

* dV/dt: The rate of change of V (membrane potential), with respect to time.
* V: the membrane potential, it usually holds a value.
* V3: A term that represents the cubic nonlinearity. It helps stabilize the membrane potential.
* W: the recovery variable, it works on restoring V to its resting state.
* I: is the external input, a positive value can encourage the neuron, a negative value can make it constant.

In this equation, there are two constants that can be changed to maintain a difference in the rate of change. W and I. Each has a different effect on the rate of change.



The second equation describes the rate of change of w in respect to the time (t). The factors that influence this rate of change are: ε, a, b, w. Epsilon is a scaling factor, and if the factors are changed, the rate of change of W will change according to the factors. Here’s what each term means.

* dW/dt: the rate of change for W (recovery variable), with respect to time.
* W: the recovery variable, it works on restoring V to its resting state.
* V: the membrane potential, it holds a value.
* Ε: Epsilon represents the timescale of W relative to V, it comes in a small positive value.
* A & B: Variables that help influence the recovery variables’ dynamics.

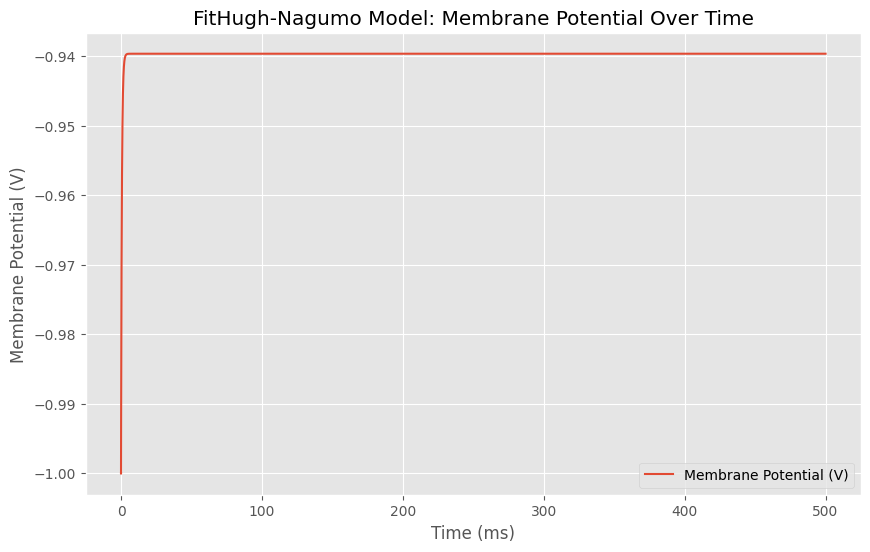
When these two equations meet, they create a second-order equation. Which will start introducing Spikes, Spikes are when the neurons start moving towards each other. This can be shown in plots inside Question 6.

Question 4:

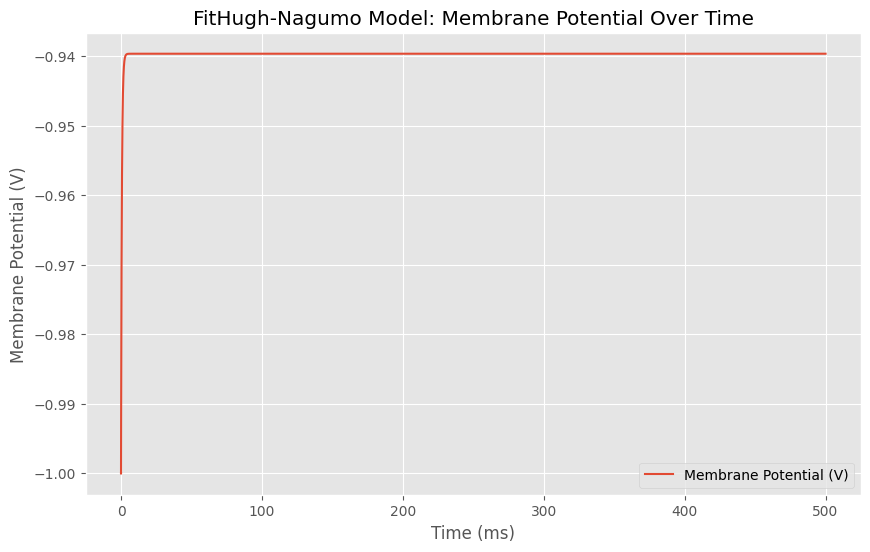
**Explain the theoretical principles of each equation and the effect of varying the differential equation's parameters.**

**First Equation.**

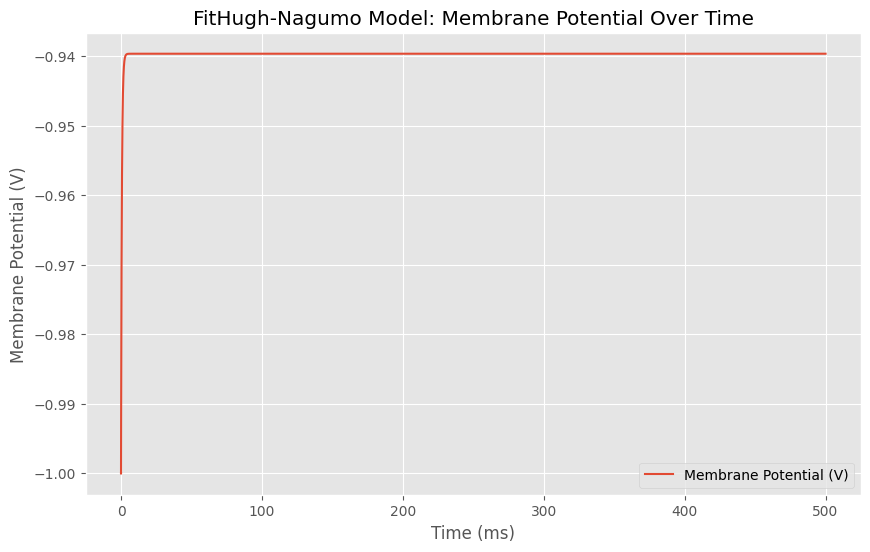
**W: Changing W in the first equation written in Question 3 while keeping I as a fixed constant, will have this effect on the membrane potential:**



W (recovery variable) is equal to 1.0 here, which started the membrane potential as -1 and then went a little up to 0.94 and then while moving across time, it went in a linear line.

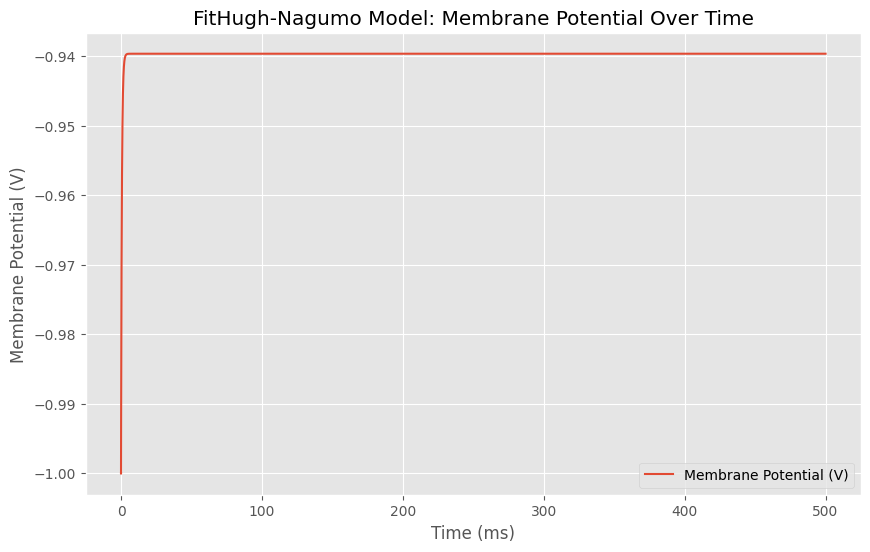


Here, we increased W to 5.0 which according to the first equation, will decrease the rate of change. Since W’s value is negative. The membrane potential also started at -1, and stayed in the same state as the previous plot. This means that increasing W didn’t change the membrane potential.

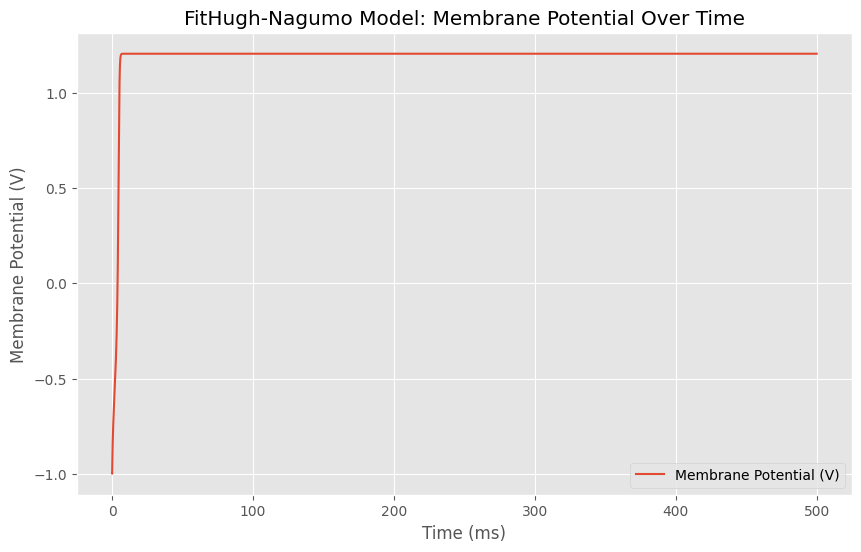


In this graph, I decreased the value of W to -5.0, this change will drastically increase the rate of change, as in the plot. The same goes for decreasing W which happened to give the same result as increasing the value. This means that W’s value doesn’t affect V (membrane potential).

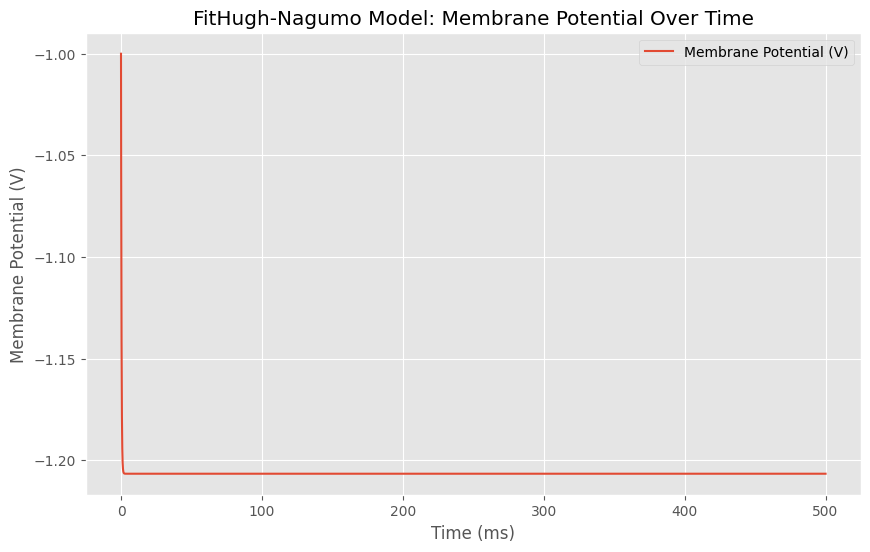
**I: Changing this constant while keeping the constant W fixed, will change the rate of change positively if I was increased, and negatively if it was decreased, it will also have an effect on the membrane potential.**



The graph at first, started with I having a value of 0.11. It started the membrane potential from the first timestamp at -1.0, then went up a little bit to nearly -0.94, to go into a linear line across the 500ms timestamps.



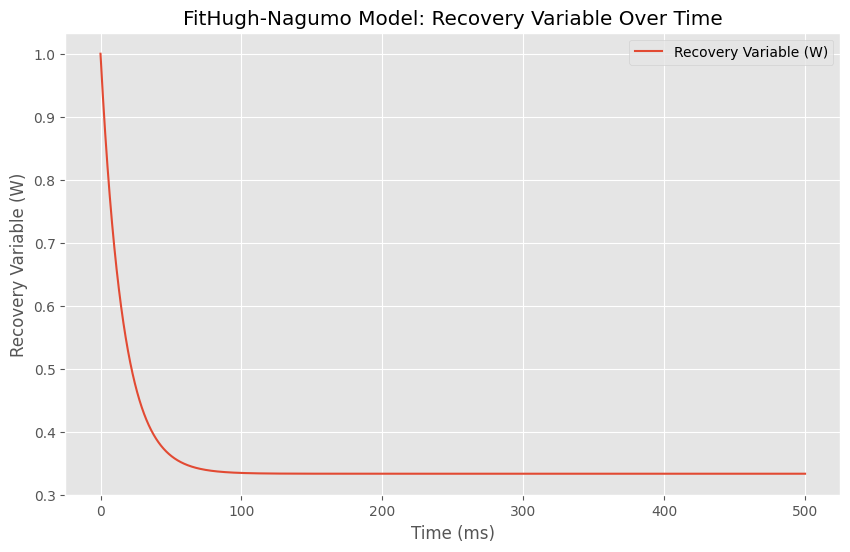
As seen from the graph, Increasing I to 0.55 changed the maximum membrane potential it goes up to before going into a linear state across the timestamps. It went up from nearly -1.0 to 1.3. Which is a drastic increase in values.



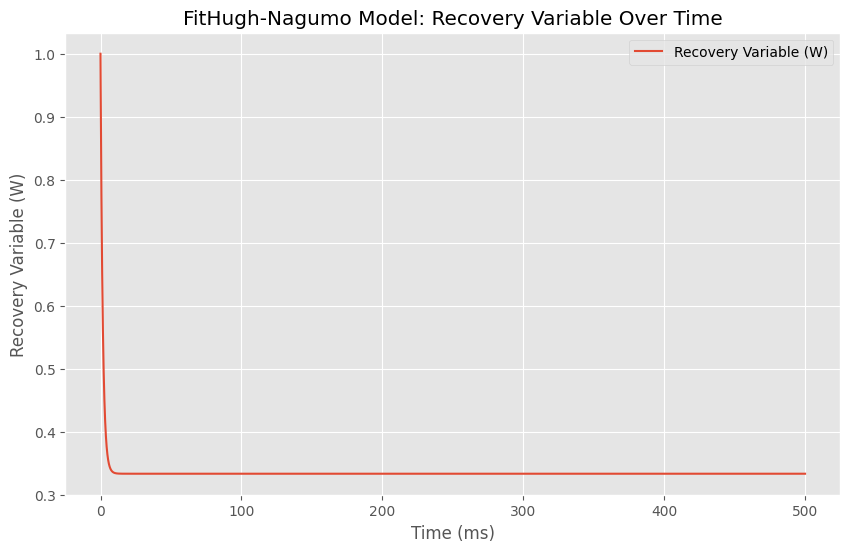
As expected, when decreasing I values from 0.11 to -0.55, the membrane potential decreased in values, going from -1.0 to -1.2. This change means that the value of I controls the membrane potential at the beginning of the timestamps. Increasing the value of I increases the membrane potential and vice versa.

**Second Equation.**

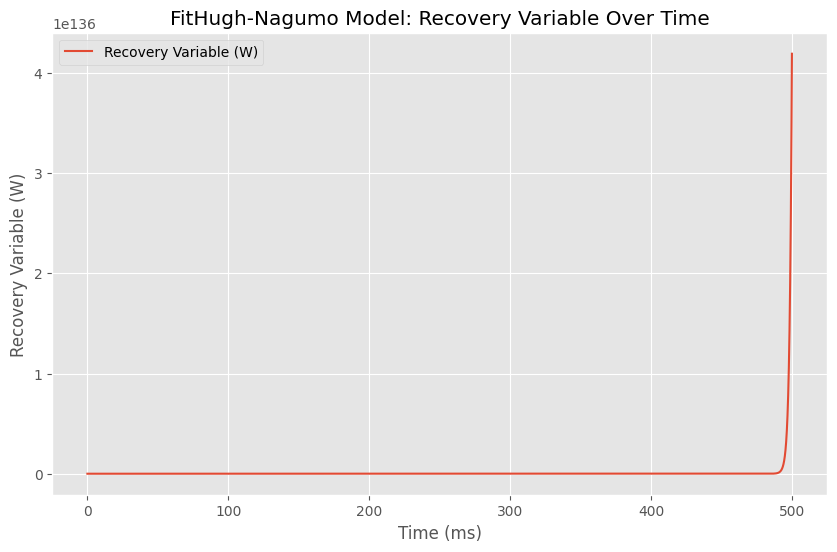
**The term ε: is the constant scaling factor that affects the rate of change of W in the speeding factor. A larger value of epsilon will grant a faster response while keeping A and B fixed.**

****

Having a and b values fixed, and epsilon equals to 0.07, gives this graph, which starts the recovery variable (W), as 1.0 then it decreases to 0.6 across time, to go in a linear line starting from nearly the 50th timestamp.

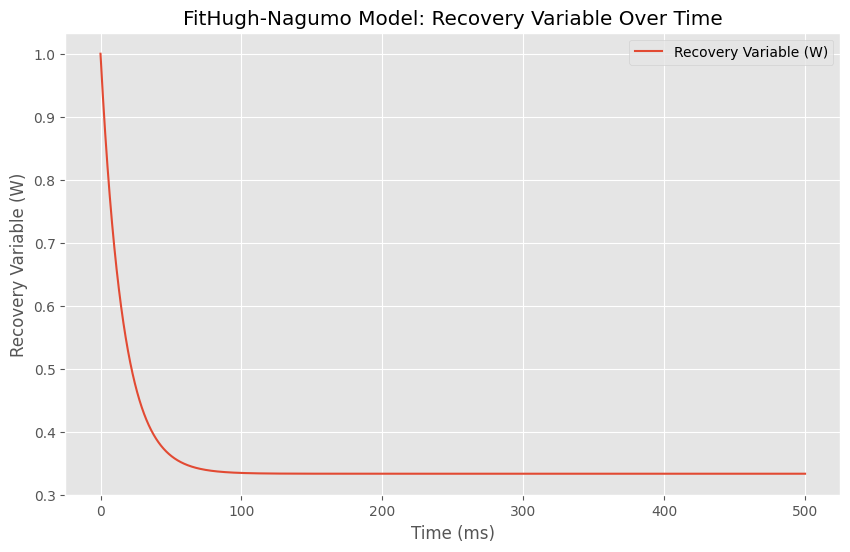


Increasing epsilon’s value from 0.07 to 0.7 starts the decrease of the recovery variable earlier than the previous plot. At nearly the first timestamp the linear line that goes the recovery variable comes across starts.

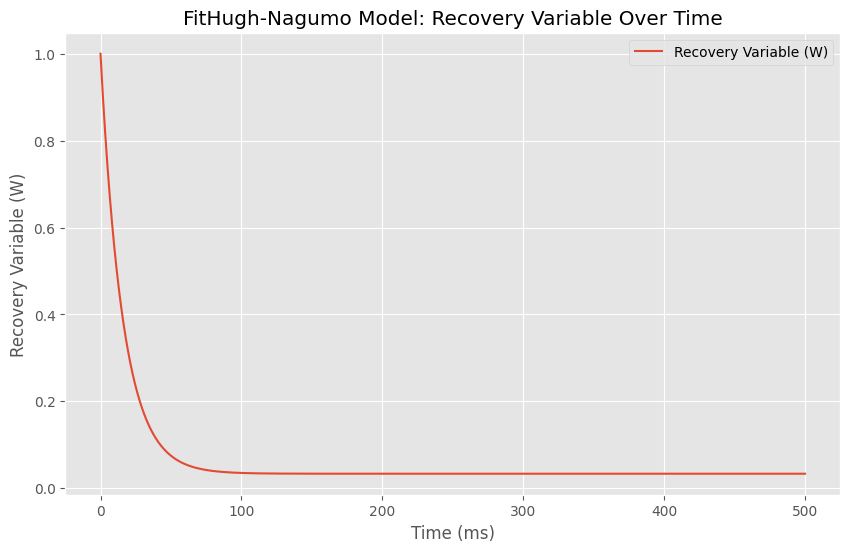


While decreasing epsilon’s value from 0.07 to -0.7 changes the increase of the recovery variable (W) largely. Since now, it starts to increase nearly at the 500th timestamp which is very late compared to the previous plots. This indicates that the epsilon constant variable controls when the recovery variable (W) changes and increases according to the time.

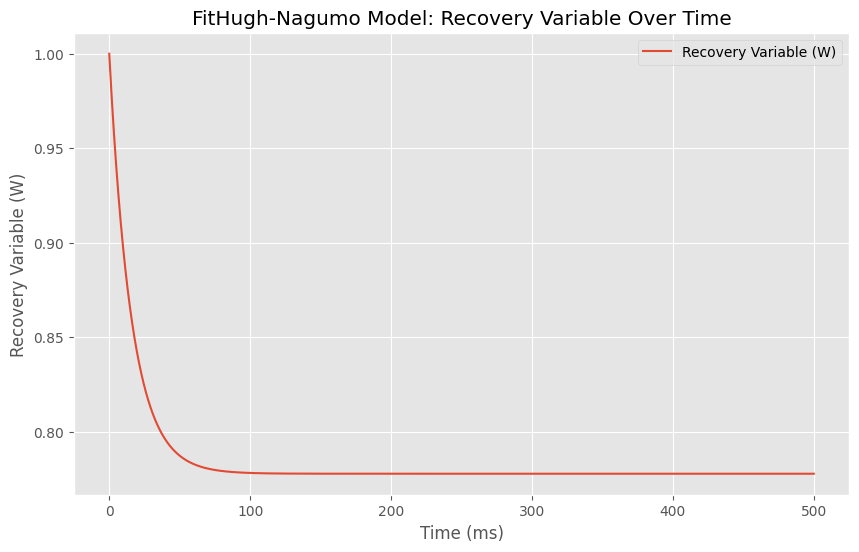
**The term a: is the term that represents the constant input that affects the baseline level of w. increasing the value of a will increase the rate of change of W, and vice versa. While keeping B and epsilon as fixed constant variables.**

****

This graph shows the normal state of the constant variable when a is equal to 0.3, it shows that the recovery variable started at decreasing at the first timestamp, until it arrived to the 50th timestamp at value 0.3, Then it started to increase in a linear line going with time.

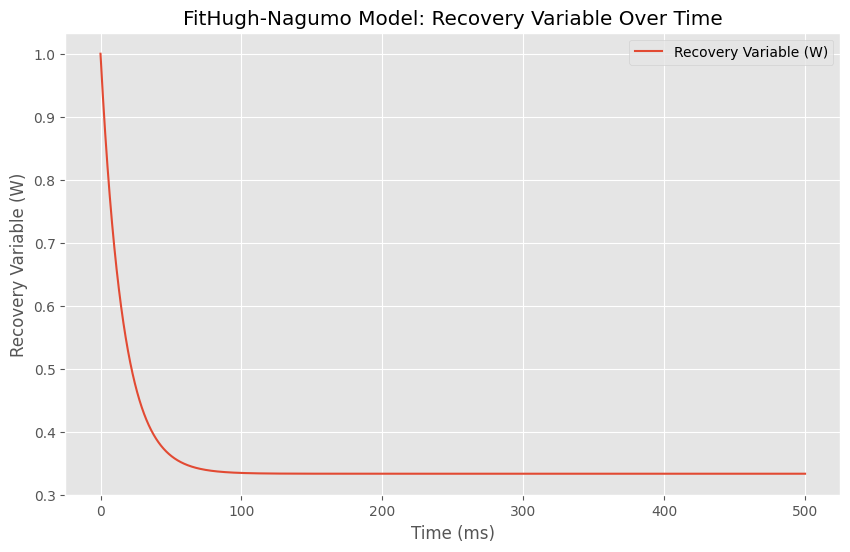


This plot shows a decrease in A as it went down from 0.3 to 0.03, it shows that the recovery variable went the same way at the start of the timestamp, but got a lower variable of nearly 0. To go into the same linear line at the start of the 50th timestamp.

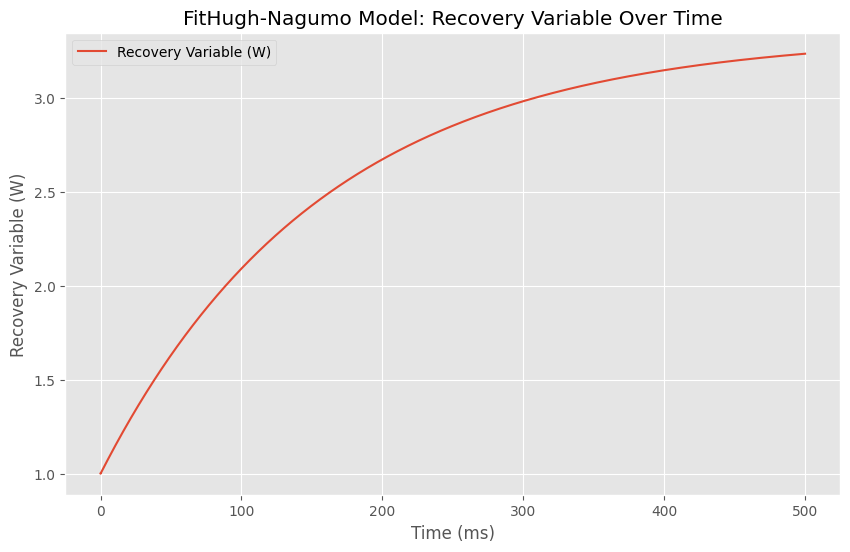


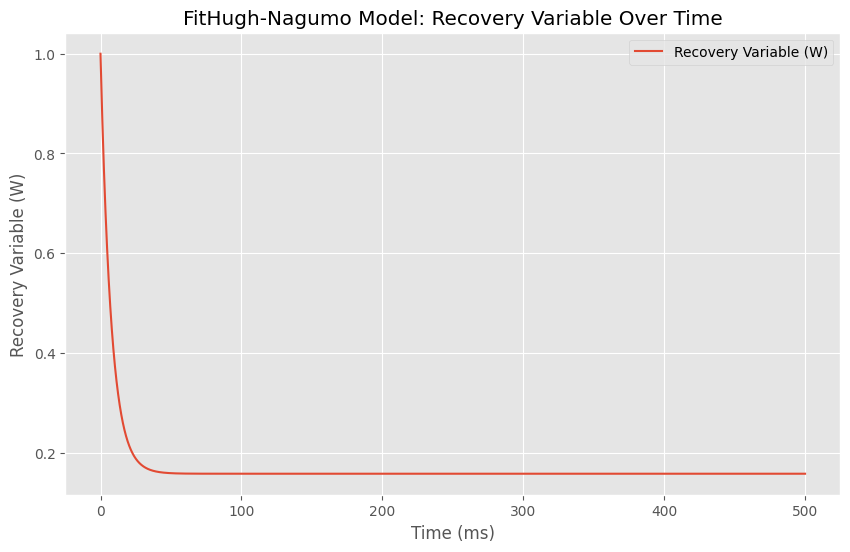
This plot shows an increase in A’s values from 0.3 to 0.7. The recovery variable (W) went down as usual but stopped lowering earlier at -0.88, before going into the same linear line at the 50th timestamp. It can also be shown that the linear line went up a little compared to the other A values.

**The term b: represents the negative feedback on W, and b is also a scaling factor just like epsilon. Increasing this value will strengthen the feedback on W, which will decrease it if V increases. While keeping A and epsilon fixed.**

****

This graph shows the normal state of the second equation, having a and epsilon fixed. And B at 0.9 shows that the recovery variable comes down to -0.6 before changing and starting to increase in a linear line at the 50th timestamp.



This plot shows a decrease in B’s values, from 0.9 to 0.09, it shows that the recovery variable (w) increased and went up in values from 1.0 to nearly 3.2 across the timestamps. Which can be considered a good performance for the recovery variable.

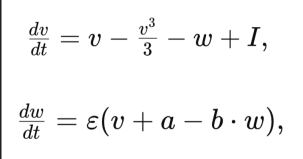
This graph shows the B value increased from 0.9 to 1.9, it shows the decrease in W a bit earlier in the timestamps. Then it goes in a straight line starting at nearly the 20th timestamp. This indicates that B controls the recovery variable’s increase and decrease across time.

Question 5:

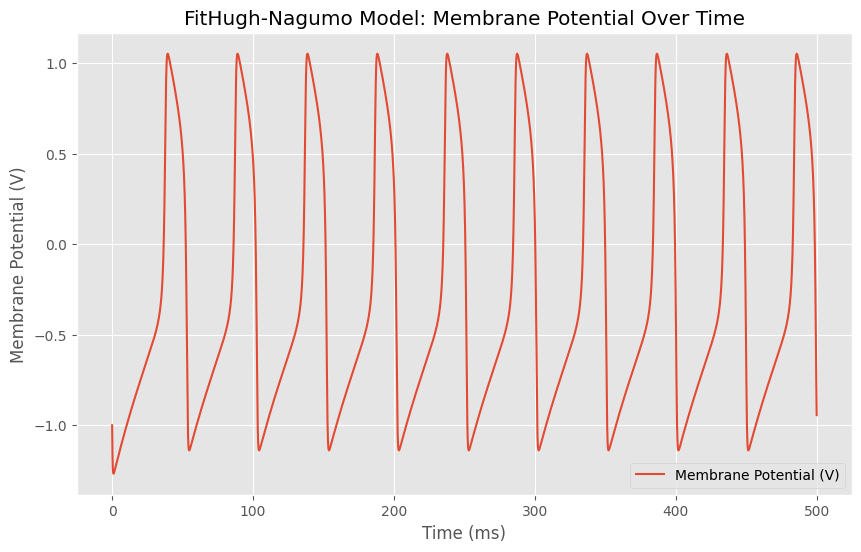
**Simulate the second-order system that consists of both equations together, where I(t) is the input, and V(t) is the output**

***The equations have been implemented in a python code.***

The second-order system will be the combination of both the equations that control the rate of change for W and the rate of change for V.



The values of A, B, and epsilon from the second equation will be changed, and plotting will start showing an increase of spikes inside the plots which indicates the use of second-order equations.



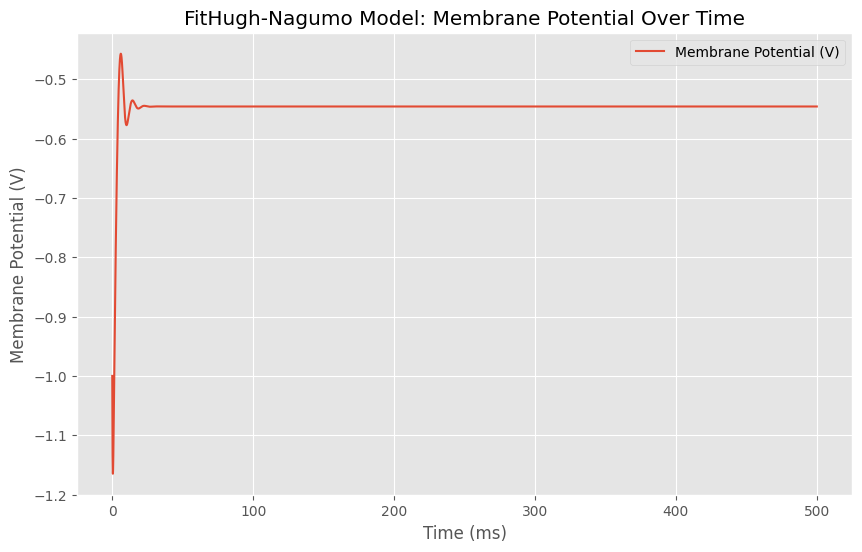
I concatenated both the W and V variables into one plot so that the change can be seen a bit clearer. This plot shows the normal state of the variables. Having A, B, and epsilon fixed. I will start changing one variable at a time to see how the plot will change.

Question 6:

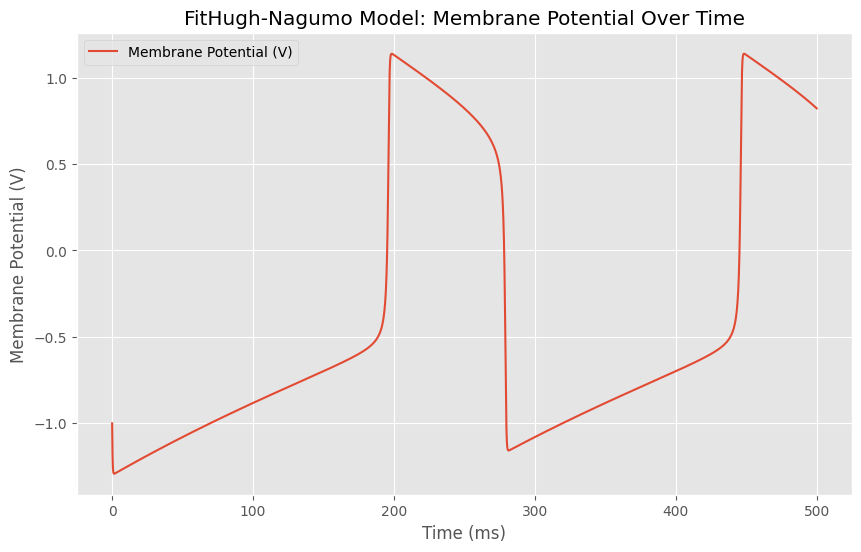
**Explain the theoretical principles of the second-order system and the effect of varying the differential equation's parameters.**

Based on the previous second-order equation, I will now start changing some of the parameters and the terms in order to find the best interpretation for the membrane potential.

Epsilon: Epsilon value started at 0.07, giving the previous plot.

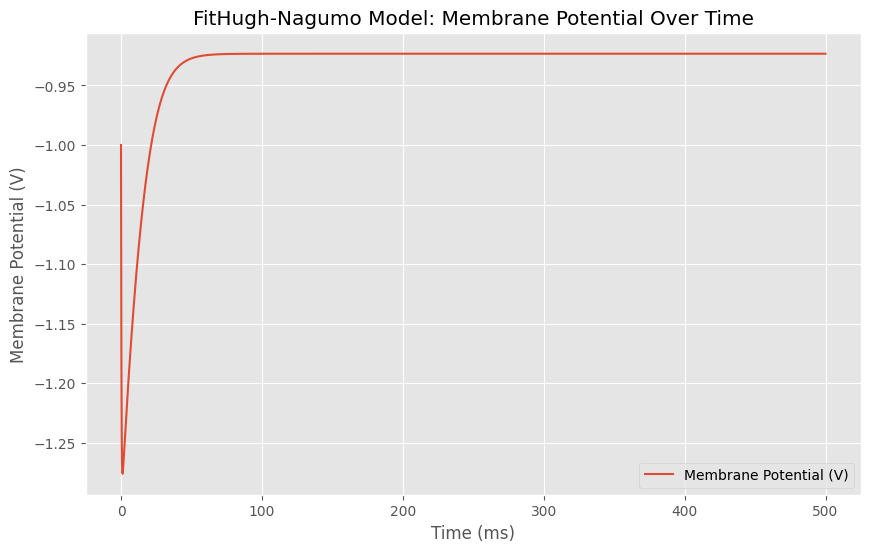


When I increased the value of epsilon from 0.07 to 0.7, the spikes had a lower chance to occur and the membrane potential stayed constant without change at nearly -0.55.

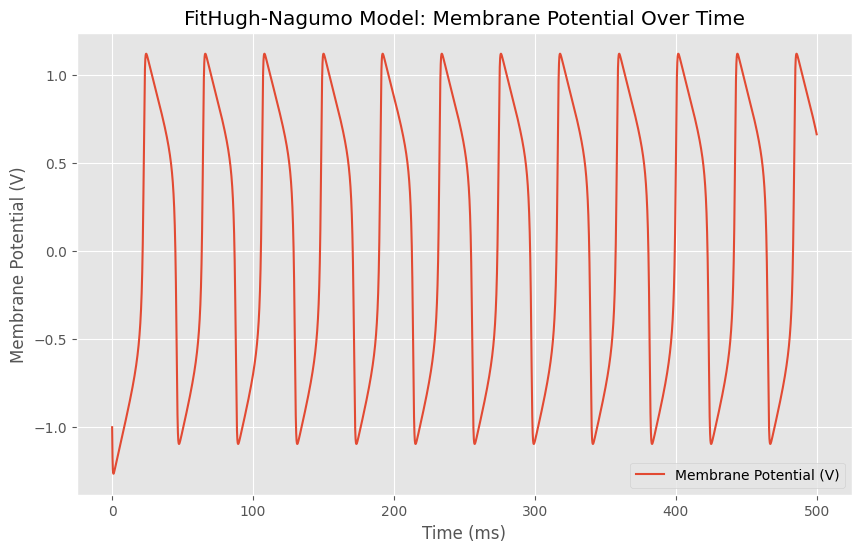


This plot shows that after decreasing the value of epsilon from 0.07 to 0.01, the spikes become very rare across timestamps (MS), but they still exist over the timestamps compared to when epsilon was increased. This shows that the epsilon value here controls the probability of spikes happening over the time.

The constant A: The value A started at 0.3 and will be decreased and increased to trace the plotting while keeping B and epsilon fixed.

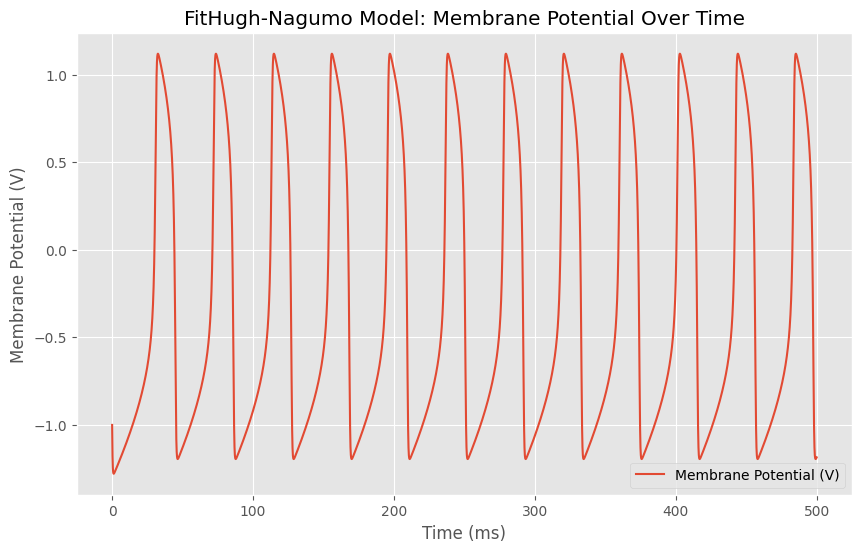


This plot shows the membrane potential when the value of A increased from 0.3 to 0.9, it can be shown that the membrane potential (v) went from -1.0 down to -1.25 and then went up towards nearly -0.9 and went in a constant line too.

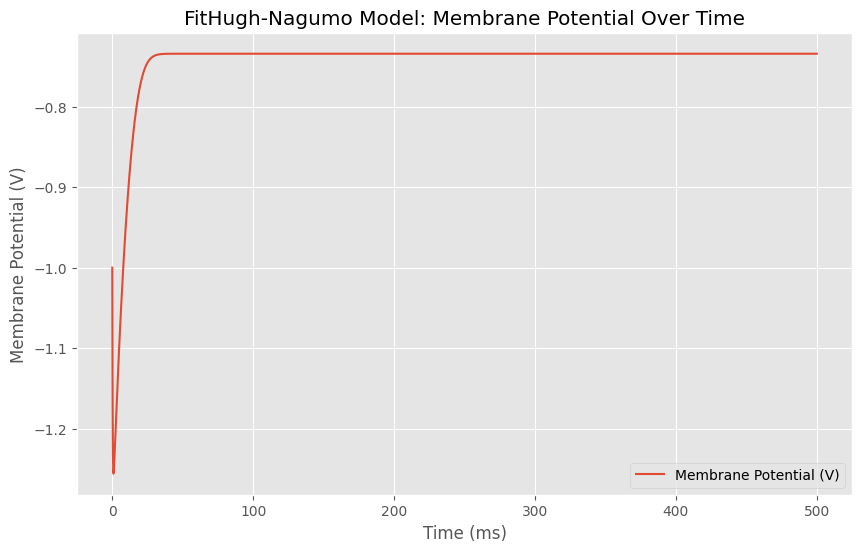


This plot showed that when A’s values are decreased from 0.3 to 0.03, the spikes become more frequent for the membrane potential. Which indicates that A’s values control the spike frequency according to the timestamps (MS).

B constant variable: The B constant variable started at 0.9, and this value will be changed positively and negatively to check how the plots change while keeping A and epsilon fixed.



When decreasing B’s values from 0.9 to 0.09, it decreased the membrane potential value lower than before, and it increased the spike frequency for the membrane potential based on the plot.



This plot was generated after increasing B values from 0.9 to 1.9, it shows the disappearance of the spikes for the membrane potential (v), and then moved in a constant line across timestamps. This indicates that B also controls the frequency of the spikes and the values of W and V. The opposite of A.

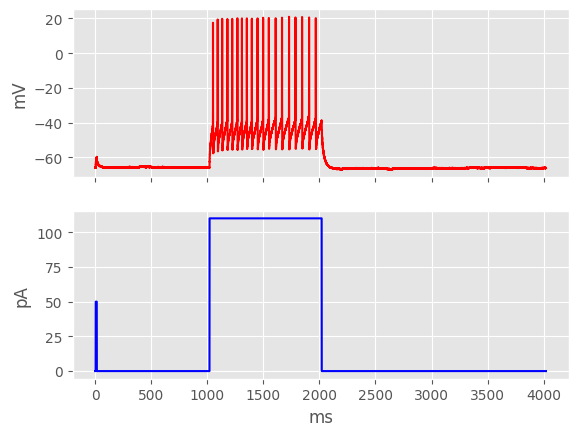
Question 7:

**Use empirical data to tune the parameters of a simulation model, the imperial data can be downloaded**

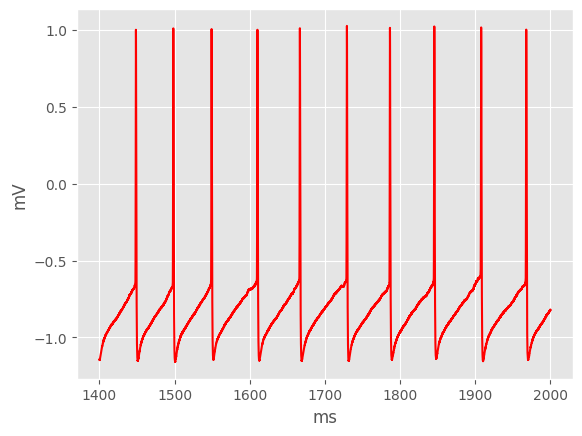
***This question is solved within the Python code attached to the final submission.***

The imperial data used was taken from the website: **https://celltypes.brainmap.org/experiment/electrophysiology/474626527**

The data used was simulated into a membrane potential equation, while I was equal to 1e12, and V was equal is 1e3. Based on those values, the following plot was given:



This plot shows the membrane potential, which was in a still state from the beginning of the timestamps until the 1000th timestamp, where it started producing spikes till the 2000th timestamp. Then it goes in a constant state again until the end of the timestamps (4000).



This plot shows a normalized version of the membrane potential in the previous plot, which focused on the spike production between the 1400th and 2000th timestamp (ms). From the graph, it can be seen that the membrane potential was increasing and decreasing from -1.0 to 1.0. going back and forth until the end of the timestamps.

Question 8:

**Apply different optimization techniques to tune simulation parameters.**

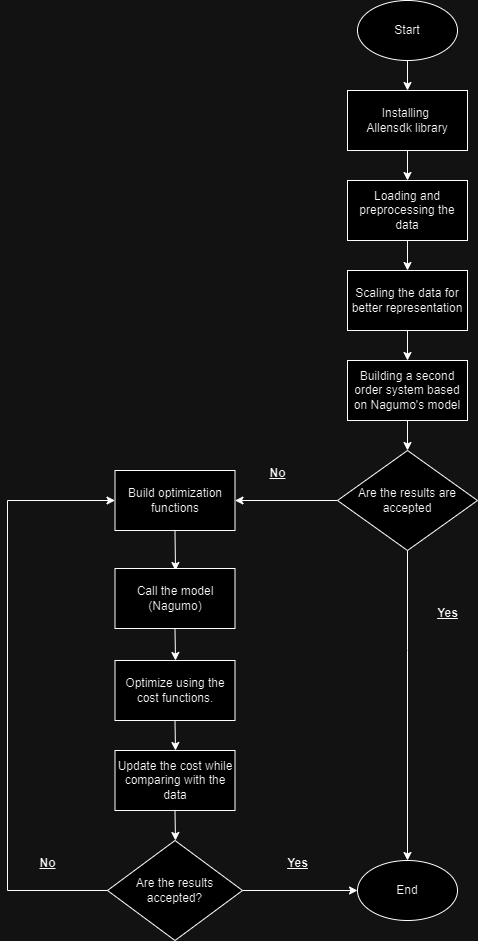
***Three different optimization techniques were used inside the code, to simulate the parameters. The Simulated Annealing, and the Gradient Descent, and the Genetic Algorithm***

Each optimization function was used on both the second-order that we created, and on the empirical data that was taken from a website. All optimization functions had a cost function, and parameters. With the best cost printed at the end.

**Task 2:**

Question 1:

**Design a detailed workflow for solving a specific problem using modeling and simulation.**

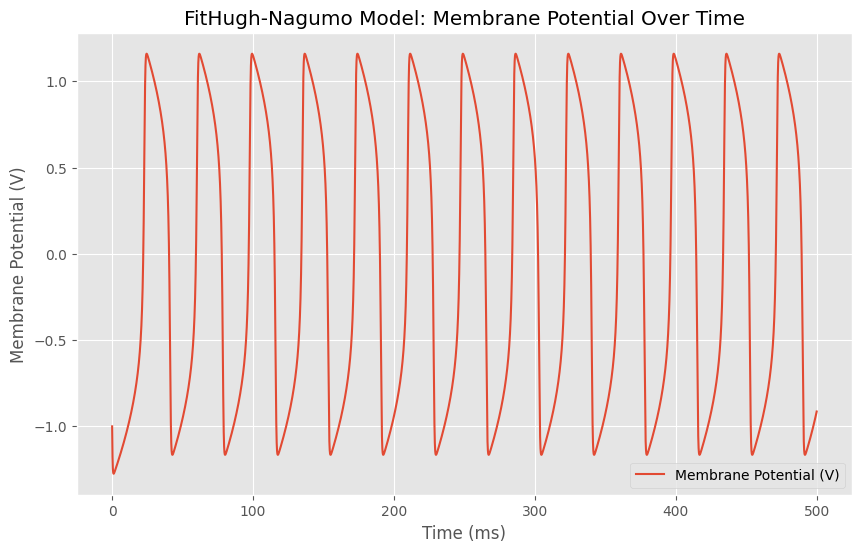
This flowchart illustrates the steps of how the code of the Nagumo model works while having three different optimization functions (Gradient Descent, Simulated Annealing, and Genetic Algorithm). Ending in accepting the model parameters and plotting as it is for the second-order system. Or updating the cost functions using the optimization functions mentioned above, and then accepting a better performance for the Nagumo model.

Question 2:

**Evaluate the performance of the second-order system in achieving desired behavior.**

To evaluate the second-order system, we need to build the system using the best parameters, given that we have tested all the parameters of the second-order system, (A, B, and epsilon), we can see that

1. Epsilon’s value was the best when it was 0.07 (because it gives the highest chance of obtaining spikes)
2. A’s value granted the best result when it was decreased from 0.3 to 0.03 So I will use it.
3. B’s value generated the best when it was decreased from 0.9 to 0.09 because it gave the most spike chances. So, I will use that value.



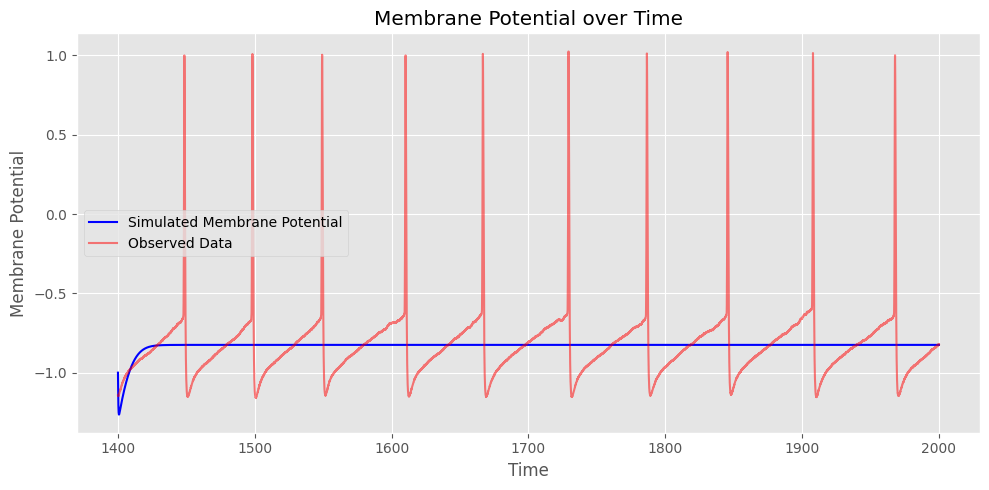
This plot was used using the best hyper-parameter tuning from the factors. (a, b and epsilon). Having using a = 0.03, b = 0.09, and epsilon 0.07. The plot generated the highest number of spike chances inside Nagumo’s model. Which is what we’re aiming for, for both the membrane potential (V). The usage of such parameters improved the spike chances that it went up from 12 spikes per 500ms to 14 spikes per 500ms.

Question 3:

**Analyze the effectiveness of each optimization technique for tuning the parameters of the simulation using empirical data.**

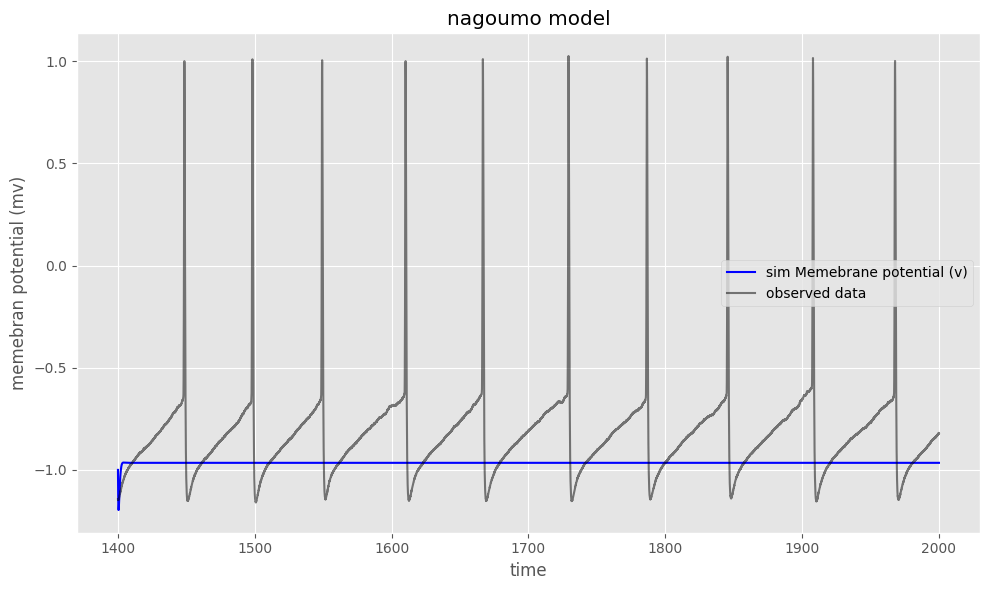
The optimization techniques used on Nagumo’s model were the simulated annealing and the gradient descent and the genetic algorithm. All were used to optimize the data and the performance of the model, based on the cost function and the tuning of the parameters. On different timestamps. Ranging from 1400 to 2000 (MS)

The first optimization function (the simulated annealing) generated this graph



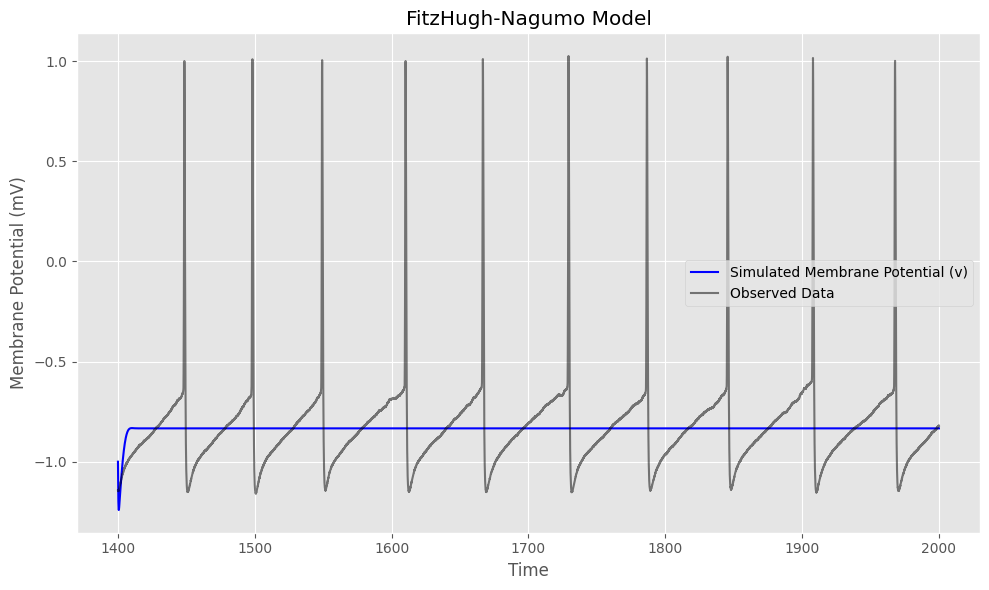
Which shows that the simulated membrane potential was not good since it was on a constant line near V = -0.8. And the observed data was different on each timestamp. This represents a bad result for the optimization function. And since the cost generated from this optimization function ranged around 1000. Which is considered a bad cost value, because that means that the error is large for this model.

Looking at the second optimization function (Gradient descent), it was trained on 1000 iterations, and the cost was printed for each 100 iterations. Ranging between the value of 1400. Which is also considered a bad cost.



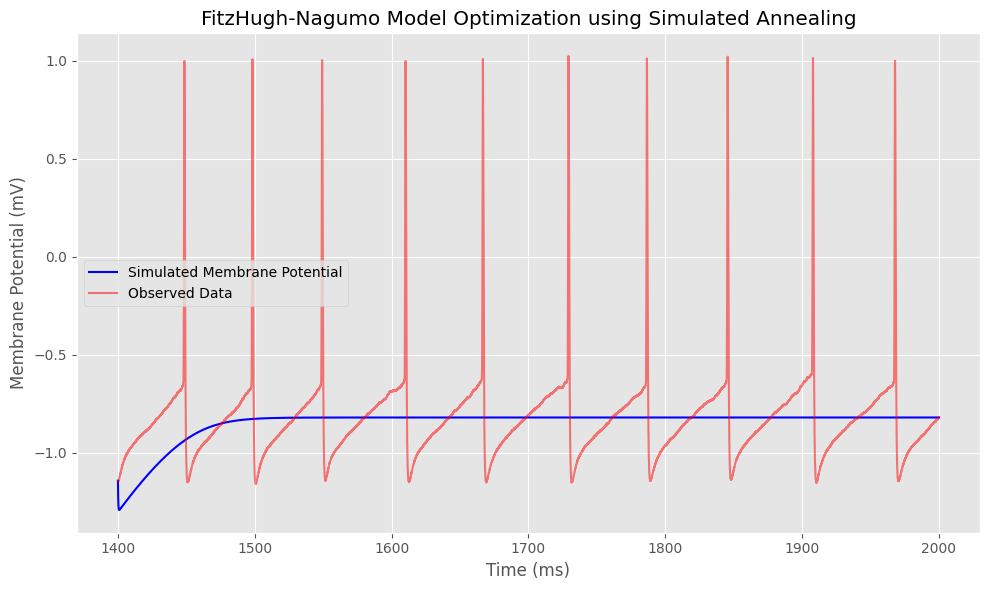
It looks from the graph that the membrane potential went into a constant line while staying on V = -1.0. This is considered worse than the simulated annealing optimization function, and the cost values were ranged between 1400.9 and 1400.5, which is a bad cost value. The observed data wasn’t read on the simulated membrane potential which is bad.

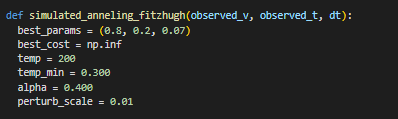
The third optimization function (Genetic algorithm) generated similar results to the previous two functions. This is the plot it generated.



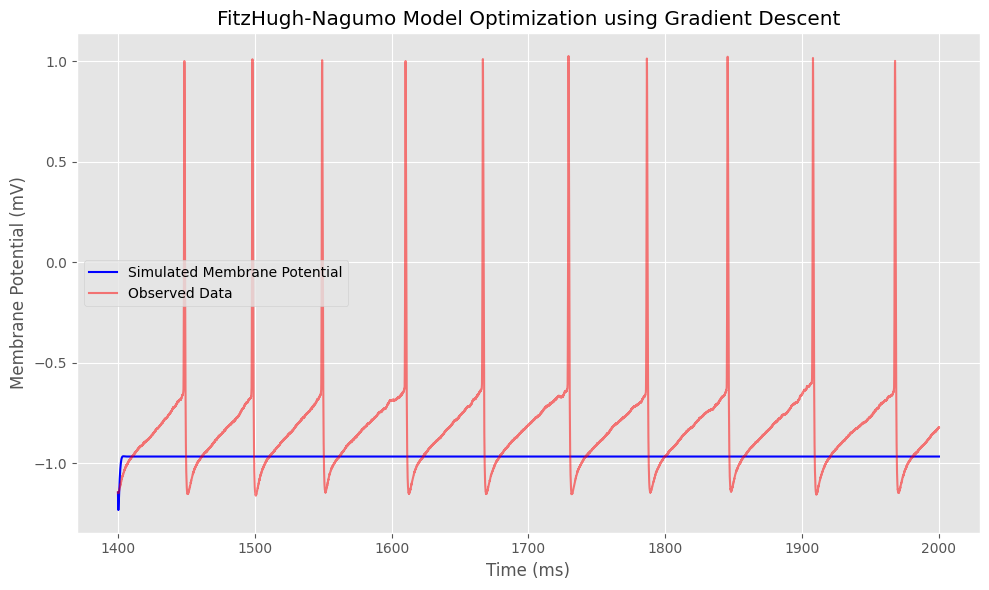
Comparing the three optimization functions (Gradient descent, Simulated annealing, and Genetic algorithm) based on the cost value and the simulated membrane potential value and its ability to read the observed data shows that the optimization function Gradient descent performed lower than the simulated annealing, since the cost value was higher (1000 and 1400), and the simulated membrane potential was lower (-0.8 and -1.0). Which indicates that the simulated annealing, even for its poor performance as an optimization function, can be used for better optimization for the values and the model of Nagumo. While the Genetic algorithm performed better than both algorithms since it went up in the value of the membrane potential earlier. And the V value rose up to -0.7, which is the highest value of the membrane potential in all three algorithms.

The three algorithms were used on the empirical data too, and generated these plots.



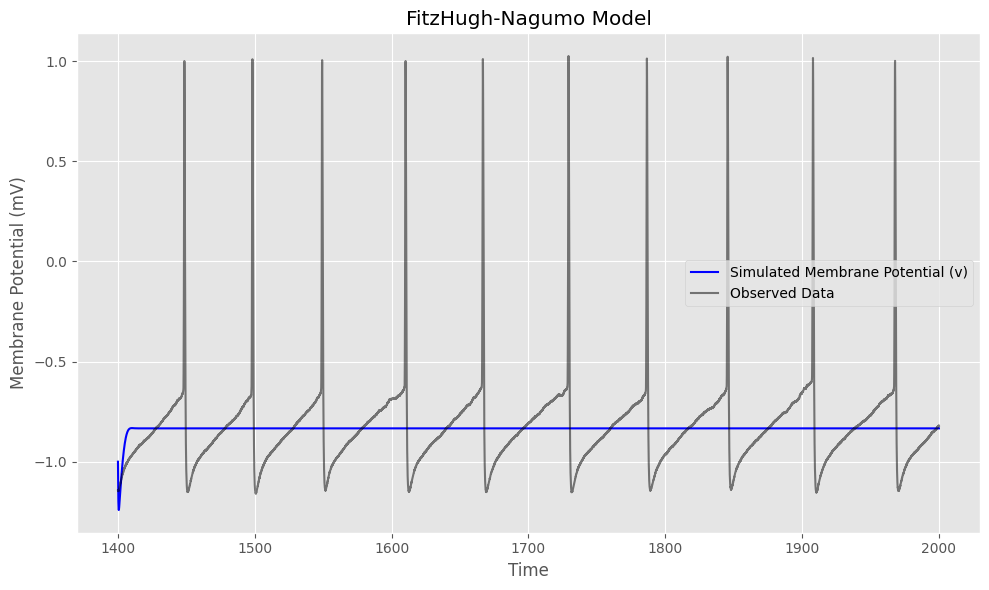


These are the parameters that were used inside the first optimization function the (Simulated annealing) it can be shown that the best params used were 0.8, 0.2, and 0.07. Which gave the previous plot as a result.

****

****

These are the parameters used for the second optimization function (Gradient Descent). It can be shown that the LR, K, V, T, and max iterations were used. The parameters used were 1.0, 1.0, and 1.0. They produced the previous plot as a result.

****

****

These are the parameters used for the final optimization function (Genetic Algorithm). It contains v, t, dt, the population size, generations, mutation rate, and the elitism rate. Based on these parameters, the previous plot was generated. With its loss.

These plots generated similar results as the previous plots, where the three optimization algorithms were used for the data. This time, the empirical data was optimized to find the best cost and the best parameters. From the plots. We can see that the worst performing algorithm was the Gradient descent. Since the membrane potential stayed at -1.0. While the other two optimization algorithms went up in value to nearly -0.75. But the Genetic algorithm reached that value before the simulated annealing. While means that the Genetic algorithm is the best performing optimization function between the three.

**Task 3:**

**Question 1:**

**Investigate previous methods and work done in the field of neuronal modeling.**

To investigate previous methods in the field of neuronal modeling, I will investigate 10 research papers that have studied this field:

1. Mathematical modeling has grown so fast in the last decade thanks for Hodgkin and Huxley’s models and simulations in the 1950s. They contributed in the neuronal dynamics, however there are vast amounts of data that weren’t combined into a mathematical model, because are unanalyzed data in the space. And developing new methods can be exhausting but helpful to make good predictions. In this paper, the ANN model was combined with optimization schemes to train input currents. This resulted in obtaining good data generations from four different neuron models when using the variables as network inputs. Predicting the empirical model is done to generalize the neuronal dynamics. The input of the model is the voltage and the current. And the results can be satisfactory, even if some predictive ability is lost. This paper contributed as a first step towards obtaining good results from empirical models by experiencing with voltage and electrical currents.
2. This paper summarized the approaches used in modeling neuronal systems in single cells, networks and other complex systems. It starts with a history about models and how neurophysiological data are combined within the AI research. Which became a good computational power source. The paper showed how integrate-and-fire models are replaced by ion channel models, and that these models do not have potential spike chances. Event-driven models that decrease the necessity to perform constant membrane-potential decay are discussed in this paper too. The paper ends in discussing some of the brain and nervous system factors that models are dependent on, while increasing the interest for future researches, and the limitations that the researchers can face and how to avoid them.
3. This paper discussed how neurons are integrated in complex networks, and that the timing of spikes is predicted by a range of ion channel currents that move across time in milliseconds to hundreds of them. The paper considered that there is a lack in analytical tools to identify these relationships, which the paper proposed a modeling approach to test the relationship between the local field potential oscillations and the timing of the spikes. Kernel density estimation is used to relate a single spike timing to a phase of LFP, then combining them within a logistic regression to show that models can test these currents. This allows researchers to test relationships between the activities and the spike dynamics overtime. In the end, the paper introduced new and evolved models that are used to study the relationships, capture, and predict the neuronal engagement in the processes. While constituting a powerful tool to test the hypothesis of spike timing overtime.
4. This paper studies some methods that are related to simulating neurons. It uses a population density and a neuronal network approach. Which results were noted, the first approach sees each neuron by its states, and it assumes a large population approximated by a continuous population density distribution. Updating this approach can show us the macroscopic behavior of such population, and the firing rate and the membrane potential too. This approach avoids the need to simulate all the neurons. And simplifies the models. The paper then introduced an APPD method that works on simulating the first approach without simplifying the model dynamics. This accesses us to various macroscopic properties of the neuronal populations as accurately as possible. It can track Gaussians by time since it is applicable to Bayesian filtering for systems. Another tracking method was used (LSKF), which finds greater accuracy in methods. The paper ends by showing how the implementation of these approaches and tracking methods are on GPU and CUDA, and that it can give real-time visualization of the neuronal activities.
5. Single-neuron modeling has been significantly useful in giving insights and predictions for complex functions, but their problem focuses on how they are built. Since they are costing for simulating living systems. But the modern methods can be trained on high-volumes of noisy data. The paper then introduces two methods to model stochastic neuronal networks. Which is a class of ANN models that provide great fixes to the high-dimensionality. Some of the dynamics are produced by a spiking neuronal network model. The ANN models can generalize better when the dataset becomes larger and larger. This covers many cases and opens the door for better modeling dynamics with ANN models.
6. This paper discussed how the creativity of humans is the key to high-performing scientific advancements. And this same creativity was studied in this paper, A proposal of some activities in the brain, called the “resting state” was mentioned, which underlies all the human creativity inside. Many experiments supported this hypothesis, and recently, a good number of modeling systems was used to understand the nature of such creativity, and the results were that, this creativity is determined by learning, expertise training, and other traits. The integration of the stochasticity led to offering original outcomes. On the other hand, some outcomes were of meaning and significance for the improvement of this study.
7. To understand how neurons in higher visual areas create relevant representations of the visual world, it's crucial to consider how they adapt dynamically to both external and internal factors. This dynamic adaptation results in a complex space, making it difficult to isolate factors' contributions to sensory information processing. The point process generalized linear model (GLM) is a widely used method for describing neuronal processing based on various inputs and linking these processes to behaviors on single-trial and individual-neuron levels. However, traditional GLMs assume static neural systems, which isn't suitable for modeling the changing sensitivity of neurons in higher visual areas. This review examines GLM variations with a focus on time-varying models, showing their understanding of neural representations and decoding neuronal sensitivity, and connecting physiological processes to behavior. These time-varying models provide critical insights into the neural mechanisms behind visual behaviors and have the potential to reveal significant computational principles across different brain regions.
8. The comparative analysis of systems of some differential equations that are ordinary. The modeling process of the gene regulatory networks and neuronal networks. This study focused on the asymptotically behavior for the solutions and how other types of attracting systems can be studied. While emphasizing on the chaotic behavior of such solutions of the differential equations.
9. This paper introduced TMS which is a stimulation that’s used to study the brain activity for brains. But it hasn’t been fully understood. Old research used modeling of an electric field to map the structure in the motor cortex. This study aimed to transfer the approach to attention. Which helped understand the TMS effect on functions. And at last optimized the schemes. As experiments went by, the results were a failed observing of the correlation between the behavioral brain and the electric field. Which introduced challenges in using such methods to check the high-neural response and complex interactions with networks. TMS selected attentional subjects and resulted in behavioral. Different patterns and results were shown while modeling using the TMS, which influenced on a significant outcome in the end. This study explored the TMS attentional modulation through the electric field for the first time, and the challenges of the finding has been highlighted for future researches.
10. Dynamic systems on networks involved processes evolving on different timestamps, in the Alzheimer disease, the spread of the toxic protein in the brain didn’t only disturb the neuronal activities, but it also increase the activity itself. Which gave a feedback loop for the fast activity and the slow spread of the toxic protein. From this disease, the paper studied multiple dynamics that work on timestamps of an adaptive network of Kuramoto oscillators, which uses two-nodes as a model. It shows the activity and the toxic outbreaks of the brain and how the spreading of the toxic protein goes. Then the model was extended on formulating larger networks and it performed many simulations on slow and fast dynamics on some common networks. These simulations studied the findings from the model. While highlighting the importance of the timestamps in the modeling of these neuro diseases.

Question 2:

**Critically analyze and compare the performance of different models used for neuronal modeling and simulation**

For this question, I have decided on comparing the Nagumo model I used in before, with a model called (Leaky integrate-and-fire), one of the papers from the previous question mentioned it and its principles. The integration phrase is called for the summation process that combines the neuronal dynamics. While combining a mechanism that triggers action potentials with some voltage, called the firing term. It is defined to be the moment that the membrane potential reaches a threshold value from below in the plot.

The leaky integrate-and-fire model is based on a differential equation:

dV/dT = (-V[t-1] – V\_rest) + R\_M \* I\_ext) / tau\_m \* dt

tau\_m   # membrane time constant (ms)

R\_m     # membrane resistance (kOhm)

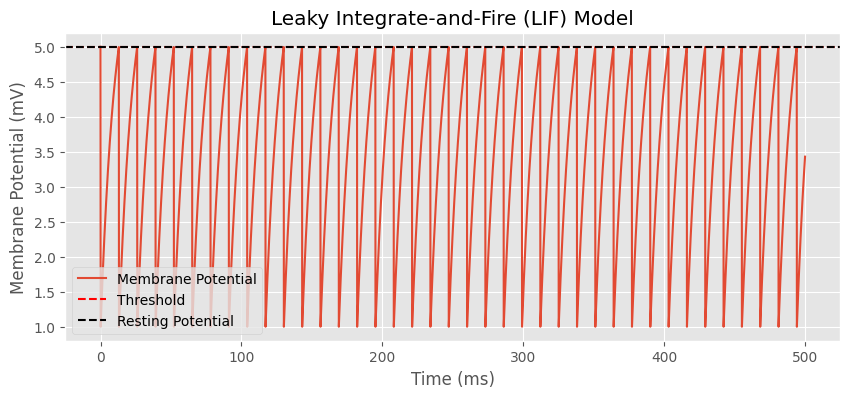
V\_th    # spike threshold (mV)

V\_reset # reset potential (mV)

V\_rest  # resting potential (mV)

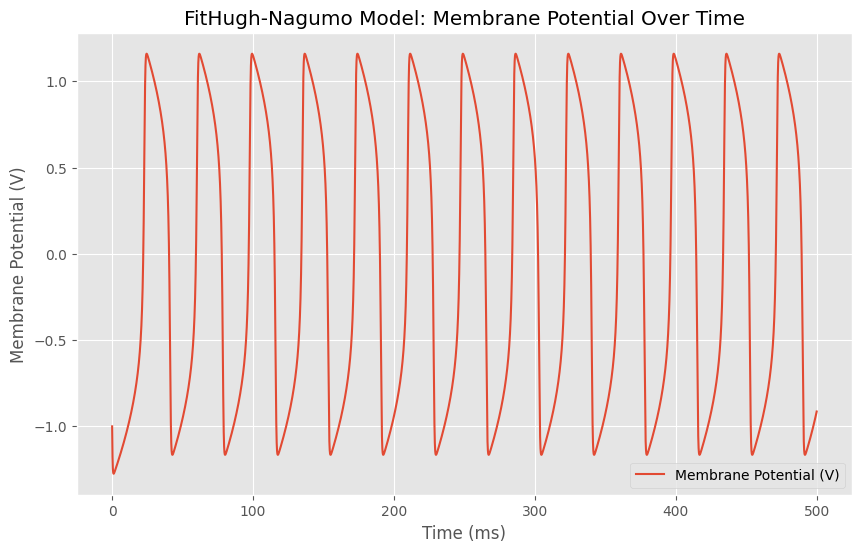
I\_ext   # external input current (nA)

These are what each parameter refers to. When implemented as a python code, and getting it to be as close to the Nagumo model’s parameters. The plot generated looked like this:



This graph shows the LIF model, which moved across 500ms, the membrane potential started from -1.0 and generated many spikes that come when the membrane potential’s value is increasing from 1.0 to 5.0 and then decreasing back to 1.0 until the end of the timestamps. The resting potential is when the membrane potential starts. And the threshold is where the neurons inside the space stay in.

Compared to the Nagumo model, which I used the best parameters for, the LIF model had a lot of spikes fired across the timestamps. Which means that the Nagumo model performed worse on the same parameters. Both models started from -1.0 and the membrane potential stayed between -1.0 and 1.0 to 5.0.



This is the Nagumo’s model plot that I was comparing the Leaky integrate and fire model to. It shows that spikes were fired, but the number generated was lower than the LIF model. This means that the LIF model outperformed the Nagumo Model in its performance

"A quantitative description of membrane current and its application to conduction and excitation in nerve" by Hodgkin and Huxley **(1952)** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/>

**"Principles of Neural Science" by Eric Kandel, James Schwartz, Thomas Jessell, et al. (2013)** <https://neurology.mhmedical.com/content.aspx?bookid=1049&sectionid=59138139>.

Pau Fisco-Compte, Aquilué-Llorens, D., Roqueiro, N., Enric Fossas and Antoni Guillamon (2024). Empirical modeling and prediction of neuronal dynamics. *Biological cybernetics*. doi:https://doi.org/10.1007/s00422-024-00986-z.

Reeke, G. (2022). Modeling Neuronal Systems. *Springer eBooks*, pp.3353–3380. doi:https://doi.org/10.1007/978-3-030-88832-9\_126.

Rivière, P.D., Schamberg, G., Coleman, T.P. and Rangel, L.M. (2022). Modeling relationships between rhythmic processes and neuronal spike timing. *Journal of Neurophysiology*, 128(3), pp.593–610. doi:https://doi.org/10.1152/jn.00423.2021.

Wang, N. (2021). Modeling and Simulation Methods of Neuronal Populations and Neuronal Networks. *Deep Blue (University of Michigan)*. doi:https://doi.org/10.7302/2885.

Zhang, R., Wang, Z., Wu, T., Cai, Y., Tao, L., Xiao, Z.-C. and Li, Y. (2024). Learning spiking neuronal networks with artificial neural networks: neural oscillations. *Journal of mathematical biology*, 88(6). doi:https://doi.org/10.1007/s00285-024-02081-0.

Malach, R. (2024). The neuronal basis of human creativity. *Frontiers in human neuroscience*, 18. doi:https://doi.org/10.3389/fnhum.2024.1367922.

Weng, G., Clark, K., Amir Akbarian, Behrad Noudoost and Nategh, N. (2024). Time-varying generalized linear models: characterizing and decoding neuronal dynamics in higher visual areas. *Frontiers in computational neuroscience*, 18. doi:https://doi.org/10.3389/fncom.2024.1273053.

Ogorelova, D. and Sadyrbaev, F. (2024). COMPARATIVE ANALYSIS OF MODELS OF GENETIC AND NEURONAL NETWORKS. *Mathematical modelling and analysis/Mathematical modeling and analysis*, 29(2), pp.277–287. doi:https://doi.org/10.3846/mma.2024.19714.

Jing, Y., Ole Numssen, Weise, K., Kalloch, B., Buchberger, L., Haueisen, J., Hartwigsen, G. and Knösche, T. (2023). Modeling the effects of transcranial magnetic stimulation on spatial attention. *Physics in Medicine and Biology*, 68(21), pp.214001–214001. doi:https://doi.org/10.1088/1361-6560/acff34.

Alexandersen, C.G., Alain Goriely and Bick, C. (2023). Neuronal activity induces symmetry breaking in neurodegenerative disease spreading. *bioRxiv (Cold Spring Harbor Laboratory)*. doi:https://doi.org/10.1101/2023.10.02.560495.

‌

‌

‌

‌

‌

‌