LSU CSC 2262, Spring

James Ghawaly

Group 12:

Austin C Louque, Grant M Candies, Jordan J Carden, Joshua R Davis, Md S Islam

INTRODUCTION

This project delves into the application of numerical methods, specifically Euler's method, to simulate neural models and synaptic transmission processes. It focuses on the implementation and analysis of Leaky Integrate-and-Fire (LIF) neurons and alpha synapses, drawing inspiration from biological systems. By employing Euler's method, we aim to gain a clearer understanding of the intricate dynamics of neural processing and synaptic behavior.

The project showcases the use of numerical methods in domains such as artificial intelligence and neuromorphic computing. Through the precise modeling of biological phenomena, we can create systems that emulate the adaptive and efficient learning processes seen in natural neural networks. This work advances our grasp of neural dynamics and paves the way for further exploration at the intersection of neuroscience and computer science.

METHODS

The project focuses on simulating Leaky Integrate-and-Fire (LIF) neurons and alpha synapses using Euler's method for numerical integration. Below are the essential components and steps involved in the simulation process.

CONFIGURATION

The project employs Python 3.10, utilizing libraries such as NumPy and Matplotlib for numerical computations and data visualization. The equations used required variables stored in a JSON configuration file. Experiments 1, 2, 3, and 5 use the following parameters shown below, with experiment 4 changing Δt to 3 ms.

$V_r = -70$ mV	$V_{thr} = -50$ mV	$V_{spike} = 40$ mV
$V_{rev} = 0$ mV	$\tau_m = 9.37$ ms	$\tau_{syn} = 0.3$ ms
$C_m = 1$ pF	$\bar{g} = 100$ nS	$t_r = 3$ ms
$w = 1$	For Euler's Method: $\Delta t = 0.001$ ms	

EXECUTION

The program takes command-line arguments to specify the simulation mode—either "spike" or "current"—and the simulation time in milliseconds. In "current" mode, the program requires an additional current input in nanoamps, while "spike" mode demands the input of the spike rate in Hz.

LEAKY INTEGRATE-AND-FIRE (LIF) NEURON MODEL

The "LIF_Neuron_Model" function simulates a LIF neuron under constant input current over a specified period, using the "LIF_Function." The "LIF_Function" calculates the new membrane voltage through Euler's method, equation shown below. The voltage is updated at each time step, considering whether the neuron is in a refractory period or has reached the spike threshold. When the voltage hits the threshold, a spike is recorded, and the voltage is reset to the neuron's resting potential.

$$V_m = V_{m-1} + \Delta t \left(-\frac{V_{m-1} - V_r}{t_m} + \frac{I_{syn}}{c_m} \right) \sigma(t - t_s - t_r)$$

ALPHA SYNAPSE MODEL

The "Alpha_Synapse_Model" function simulates a LIF neuron receiving input spikes at a provided input spike rate. Using the "generateSpikes" function, input spikes are generated throughout the simulation period based on the spike rate and simulation time. The synaptic current is calculated using the equation shown below. This calculated synaptic current is then used to update the membrane voltage with the "LIF_Function" at each time step.

$$I_{syn} = w\bar{g}(V_{rev} - V_m)\frac{t-t_0}{\tau_{syn}}\exp\left(\frac{-(t-t_0)}{\tau_{syn}}\right)$$

VISUALIZATION

After completing the simulations, membrane voltage over time is plotted for both LIF neuron and alpha synapse simulations. This visualization provides valuable insights into the behavior of the neuron and synaptic transmission processes.

CHALLENGES

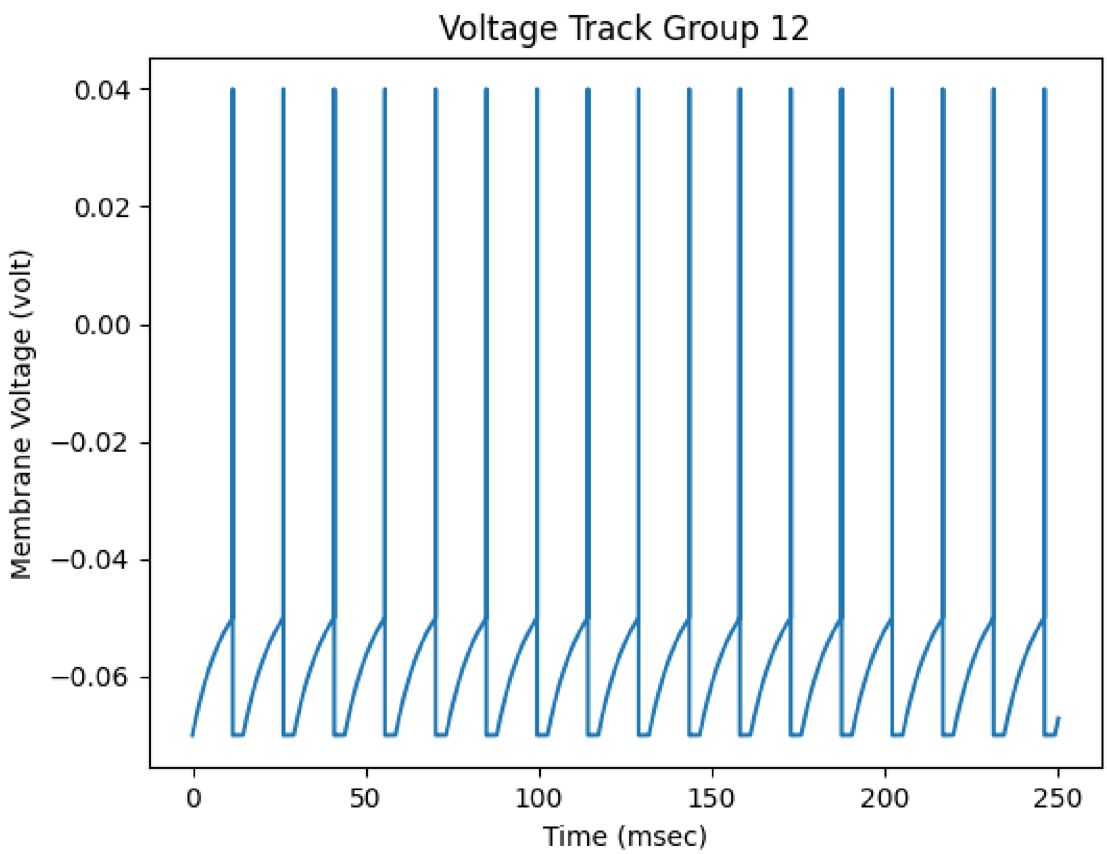
The project encountered round-off errors when generating an array of time values using "np.arange" from zero to the simulation time. This issue arose because all time values in the JSON configuration file were multiplied by "E-3" to represent milliseconds, necessitating the division of simulation time by 1,000. Consequently, the array created from "np.arange" had a very small step size, resulting in round-off errors.

To address this, all time values from the JSON configuration file were multiplied by 1,000 and the simulation time was kept as provided. Additionally, to ensure correct times and avoid further round-off errors, the program uses a function to count the number of decimal places in "deltaTime" and all time values are rounded to match the number of decimal places in "deltaTime."

RESULTS

Experiment 1:

python neuro_sim.py current 250 --current 0.003

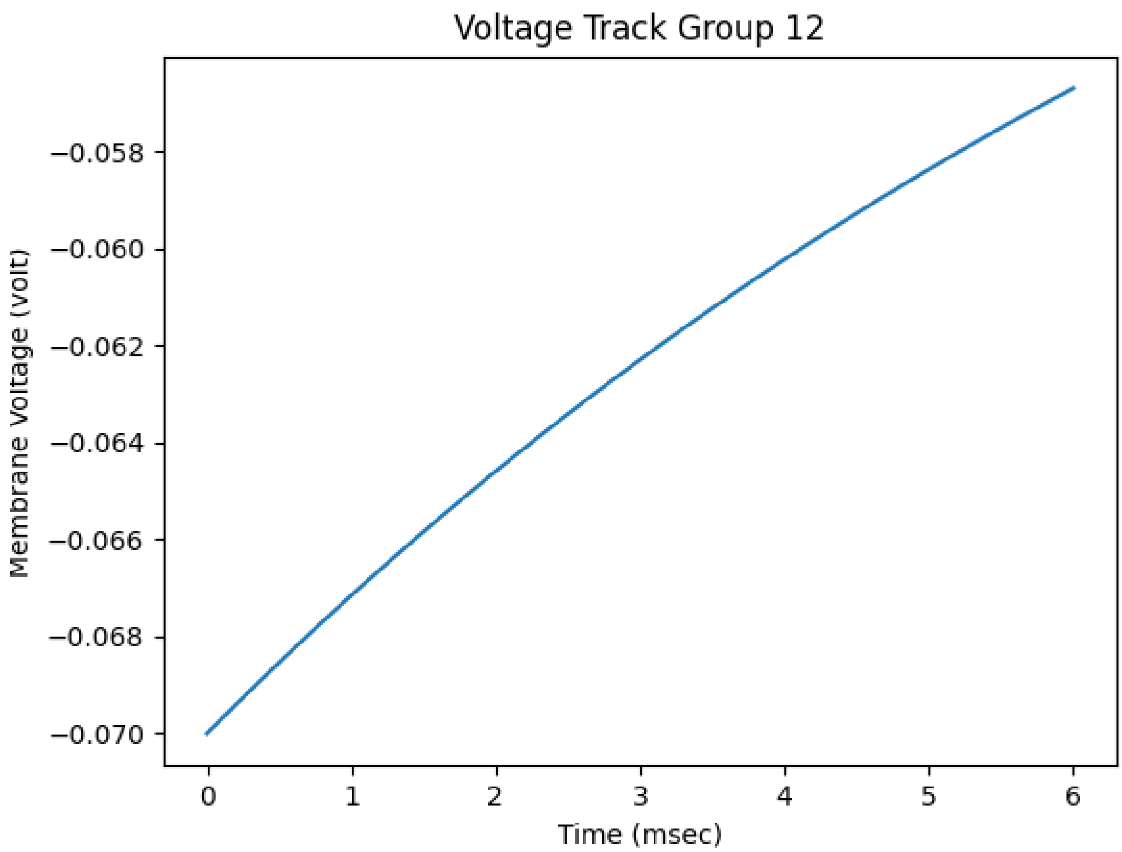


Q2: How many spikes were produced over the course of the experiment?

17 spikes

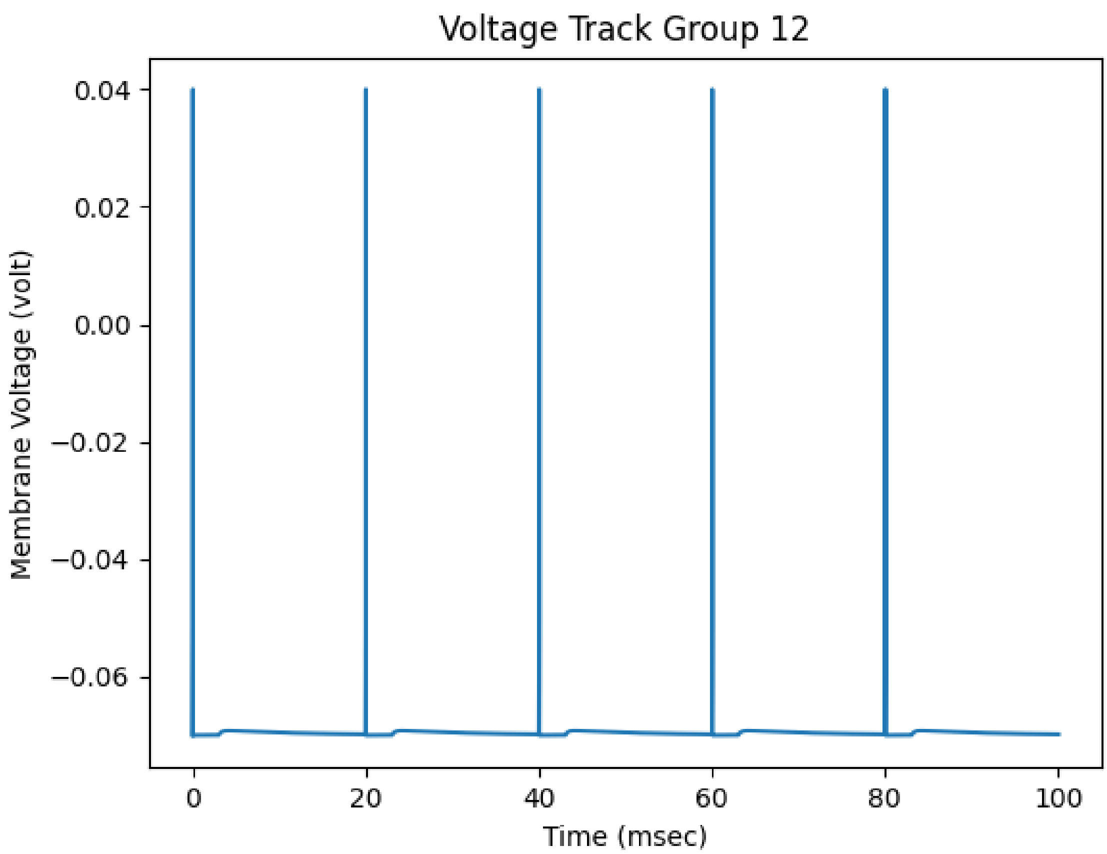
<u>Experiment 2:</u>

python neuro_sim.py current 6 --current 0.003



Experiment 3:

python neuro_sim.py spike 100 --spike_rate 50

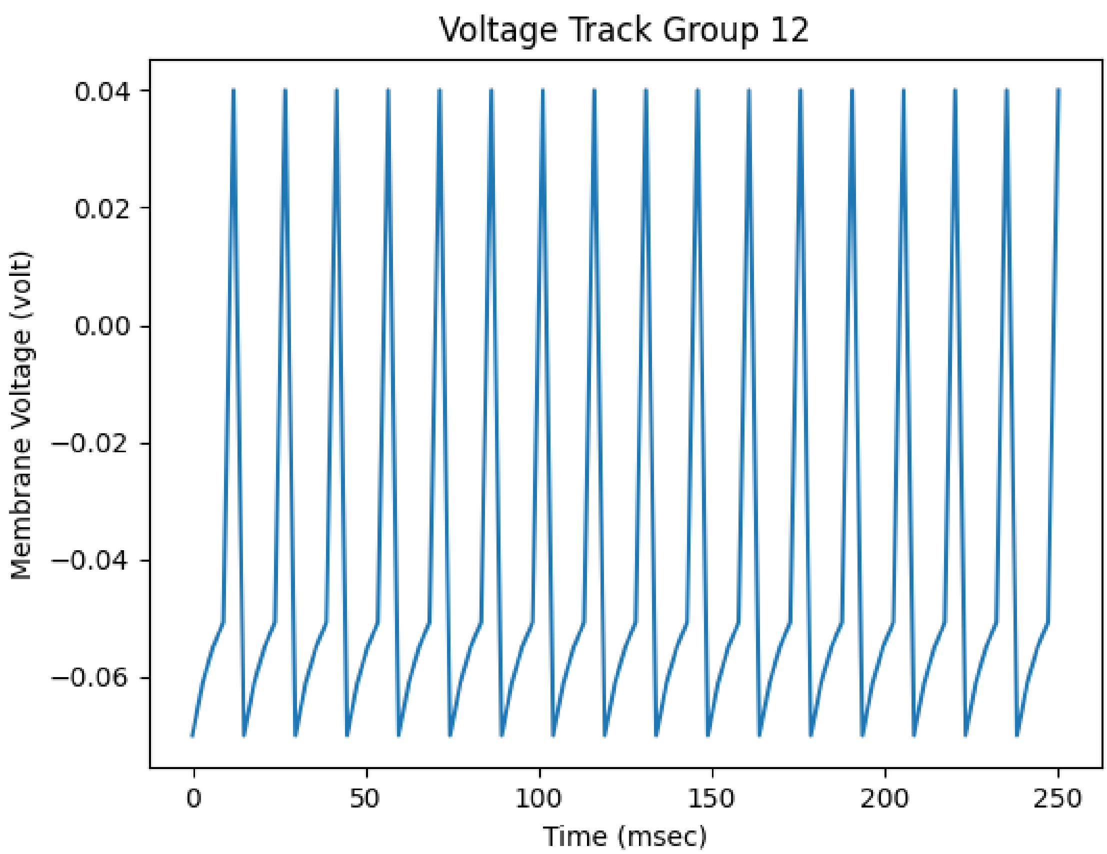


Q2: How many spikes were produced over the course of the experiment?

5 spikes

Experiment 4:

python neuro_sim.py current 250 --current 0.003



Q2: How many spikes were produced over the course of the experiment?

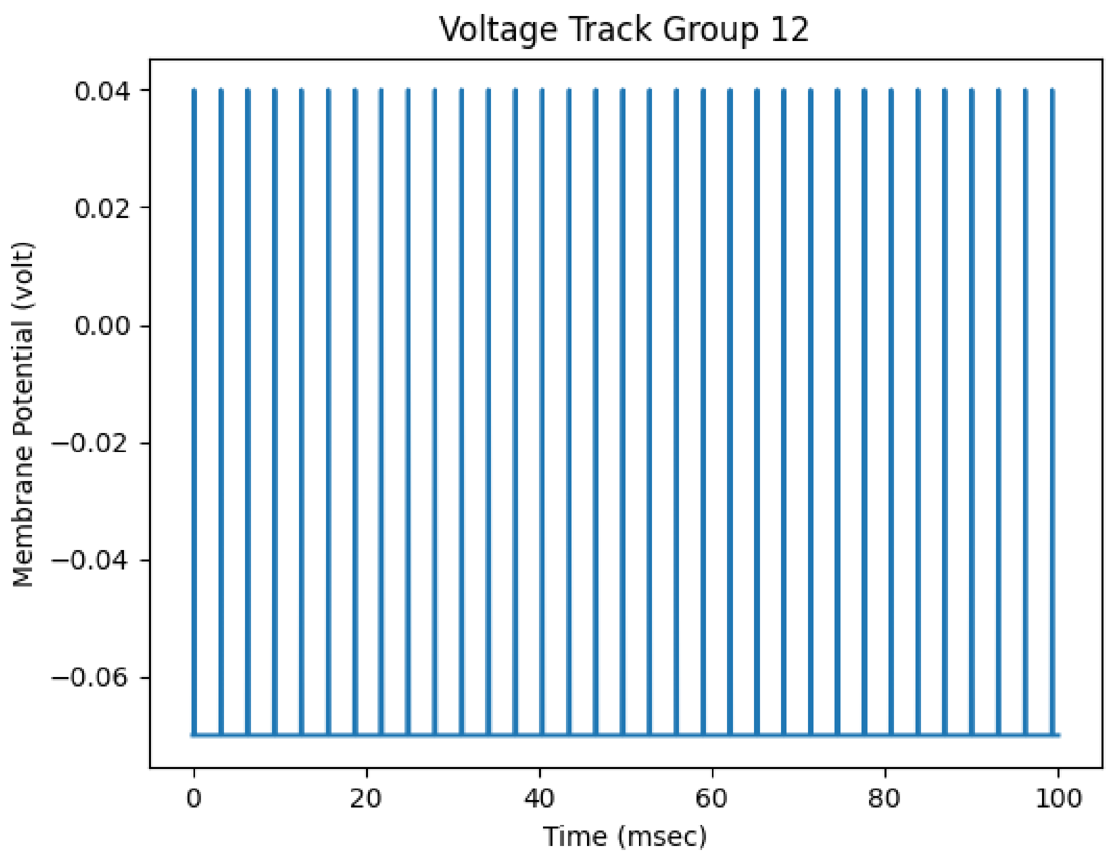
17 spikes

Q3: How does this compare to the result from Experiment 1?

Due to the larger step size, the graph exhibits reduced accuracy, resulting in a greater number of interpolated points or "guesses" made by the graph.

Experiment 5:

Repeat experiment 3, but replace the exponential term, $\exp\left(\frac{-(t-t_0)}{\tau_{syn}}\right)$, with a 10th order Taylor Series approximation of ***e***^{x} centered at 0.



Q2: How many spikes were produced over the course of the experiment?

33 spikes

Q3: How does this compare to your result with Experiment 3?

While the results for experiments 5 and 3 share the same format, the results of experiment 5 display a significant increase in the number of spikes comparatively. The taylor series approximation of e^x incorrectly calculates the voltage to be greater than the spike threshold at an interval equal to the refractory period.

CONCLUSION

We simulated Leaky Integrate-and-Fire (LIF) neurons and alpha synapses using Euler's method as a means of numerical integration. By leveraging command-line arguments to customize inputs such as simulation mode, time, spike rate, and current, we demonstrated how various excitation methods influence neuronal behavior. Through the simulation, we investigated how neurons respond to different patterns of input and observed the corresponding shifts in membrane potential and spike patterns over time. The outcomes demonstrated the efficacy of Euler's method in simulating neuronal dynamics, underscoring its potential for modeling basic neural networks.