Brian Wang

113355615

Project # 4

Resources Used: Matlab documentation, Lecture PDFs

I consulted with my group members Aaron Das, Amrita Shah, Xin'er Jiang, Kevin Hazelton on small details such as what to name the graphs and issues with the MATLAB code. All the code was done by me and I received no extra help in making them.

By signing below, I attest that the statement made above represent a complete account of the materials I used in completing this assignment. I understand that the failure to disclose the use of any resource is an act of academic dishonesty subject to penalty by the academic judiciary

Signature: Brian Wang

Date: 4/15/23

1. (Main Simulation)

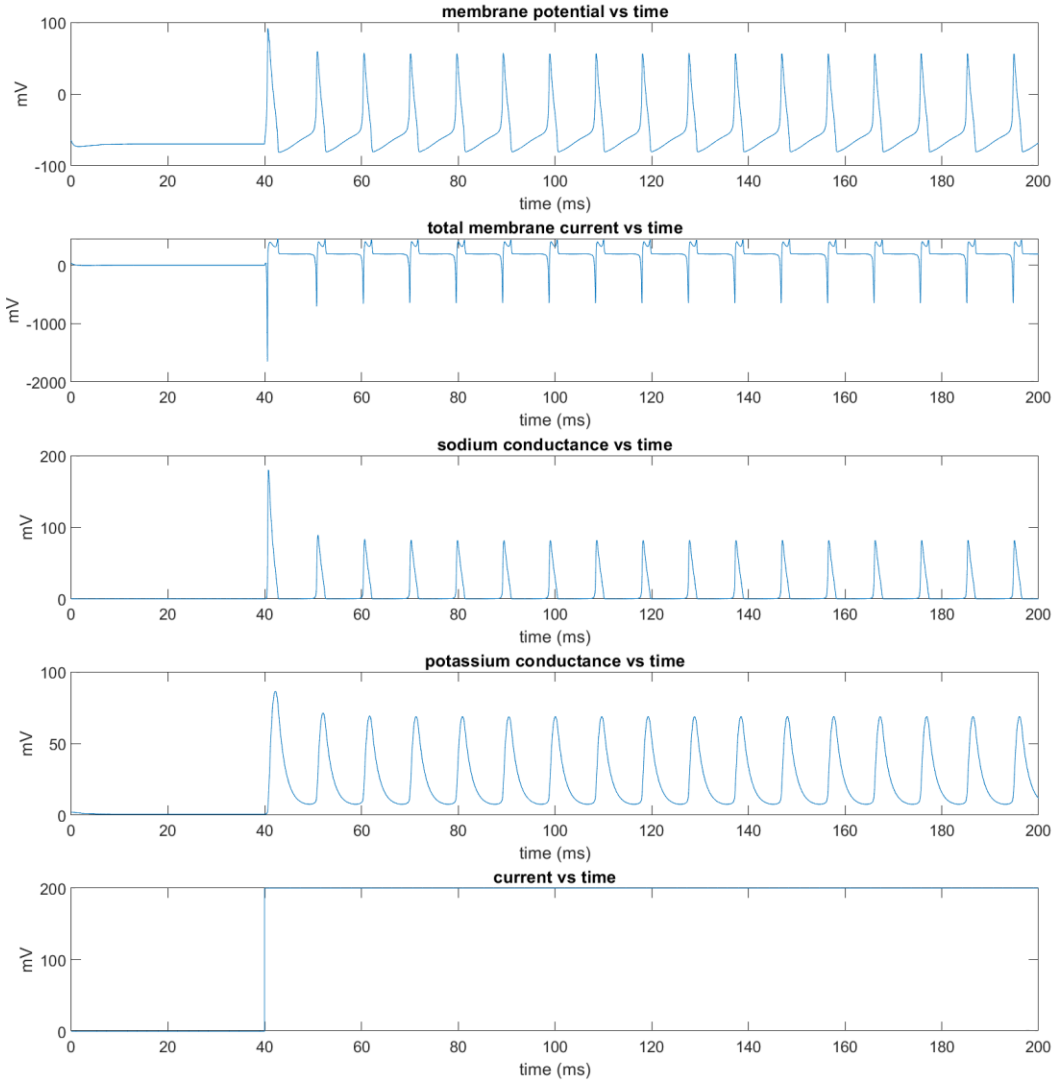


Figure A1 graphs containing data for membrane potential, total membrane current, sodium conductance, potassium conductance, and current. All are graphed against time. The pulses only started after the current increased to 200, suggesting that this has a major impact on whether the neuron fires or not.

2. (Spike Detection)

```
if V(i + 1) > detecThreshold
    if spikeRecorded == false;
        totalSpikes = totalSpikes + 1;
        spikeRecorded = true;
    end
else
    spikeRecorded = false;
end
```

Figure PartA2 In order to implement this piece of code, a variable was used to represent whether or not the spike has been recorded or not, a new spike will be recorded if and only if the spike detected variable is set to false and the threshold has been reached. If the threshold has been reached but the spike has already been recorded, then nothing else would happen. In MATLAB, this code will look like this and the total number of spikes that has been recorded is 17 spikes.

3. (Action Potential Threshold)

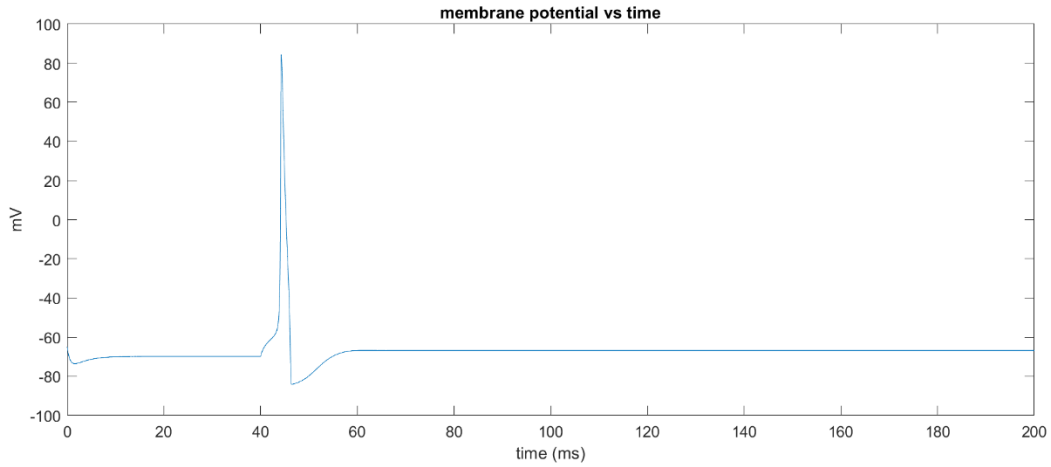


Figure PartA3 with a current of 19 mV, a single action potential is able to be fired. There is a large range beyond 19 mV that would also result in a single action potential being fired. A current of 18 mV will result in an incomplete action potential, ramping up but not having sufficient velocity to depolarize.

4. (Rheobase current)

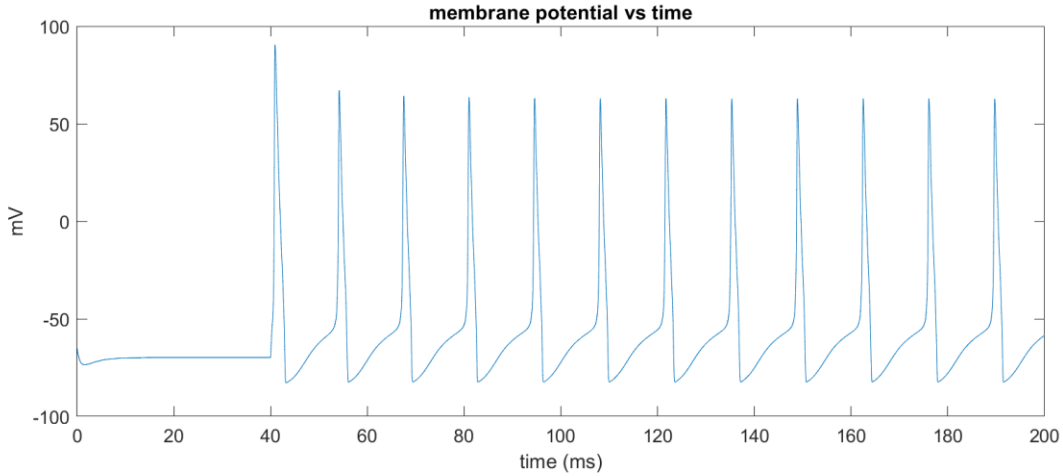


Figure PartA4 A current of 110 mV is required in order for repetitive firing to occur. A higher value will result in repetitive firing but at a lower interval. A lower value will result in limited firing to occur.

5. $(f - I\ curve)$

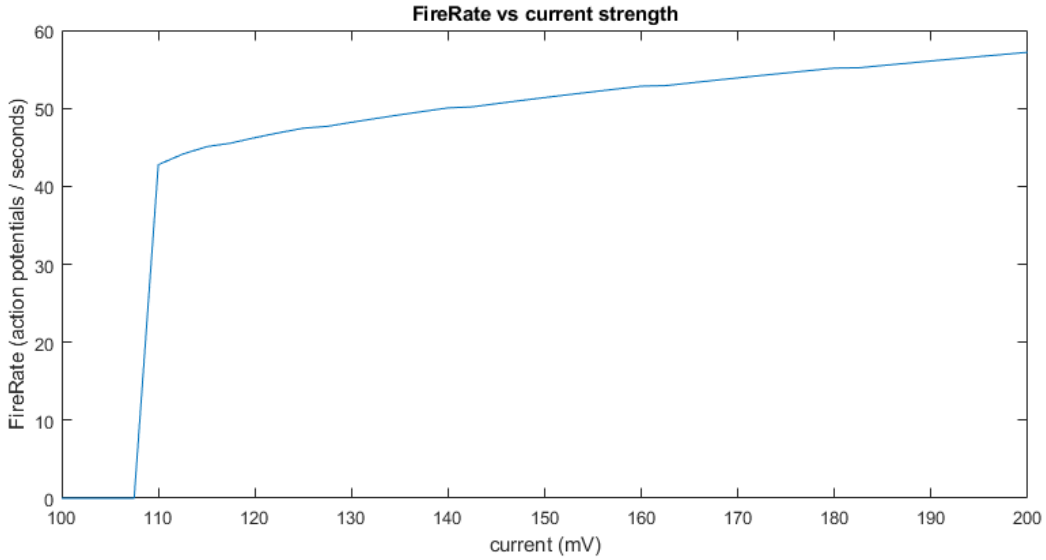


Figure PartA5 The change in fire rate over time. Generally as the current increases, the fire rate measured in action potentials per second increases. This is likely due to the added current that increases the rate at which depolarization happens.

i. the F-I curve does not have a fire rate that starts from 0 and increases onwards. Instead by the time the system registers that there is continuous repetition, the fire rate already starts at around 40 action potentials / second. Another thing to note is that that the rate of change in the fire rate seems to decrease as the current increases, suggesting an upper limit for fire rate

ii. I_{rh} seems to be at around 110 mV

iii. Yes, since we would get a good idea of the output fire rate.

Part B: Leaky Integrate-and-Fire neuron Model

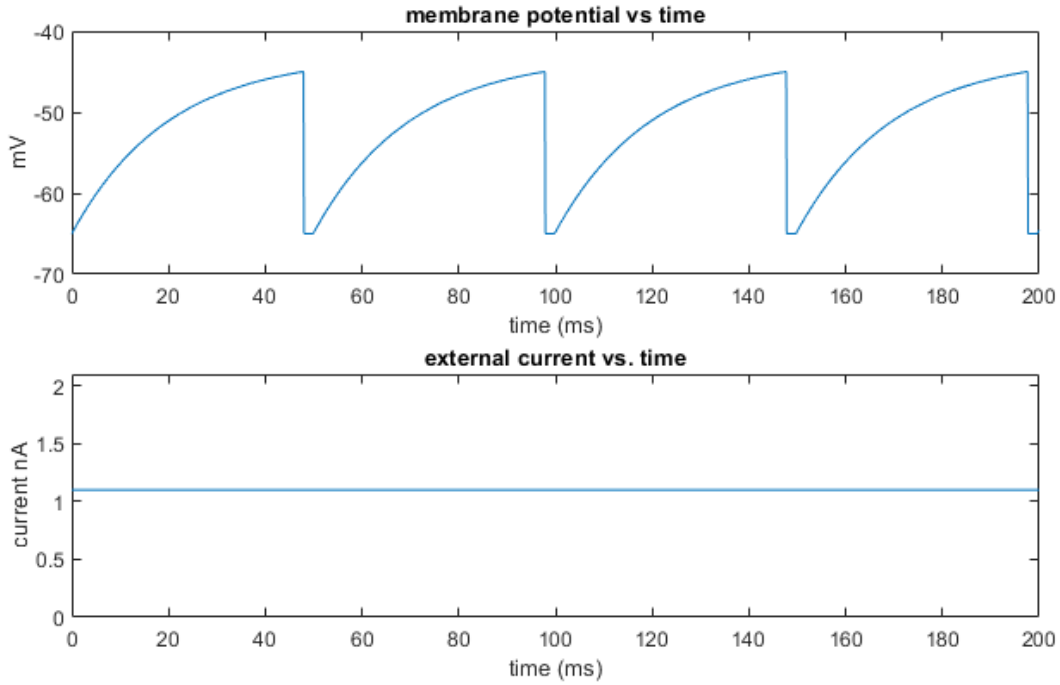


Figure PartA1 the membrane potential overtime. It looks much more different compared to the HH model due to the simplified nature of the equation for the change in membrane potential over time. The horizontal sections is the absolute refractory period in which $\frac{dV}{dt}$ has no effect on the membrane potential.

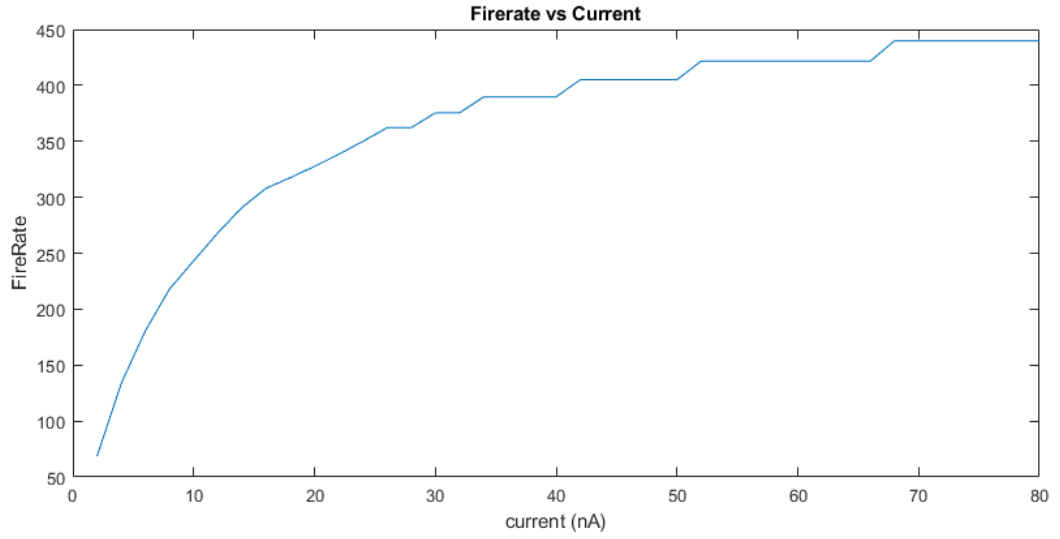


Figure PartA2 The (F – I curve) for this model. The rate of change in the fire rate decreases as the current increases suggesting a maximum fire rate for this model.

i. The curve increases quickly and then levels off. This is different than the curve for the HH model where it has a high initial fire rate but the rate of change in the fire rate does not change significantly

ii. No, the model is not complex enough to support something like that. This model can only have two states. Continuous firing or no firing at all dependent on the input current.

iii. The rheobase current cannot be estimated from this graph. However the theoretical rheobase current would be $G_L(V_{spk} - V_L) = 1000$.

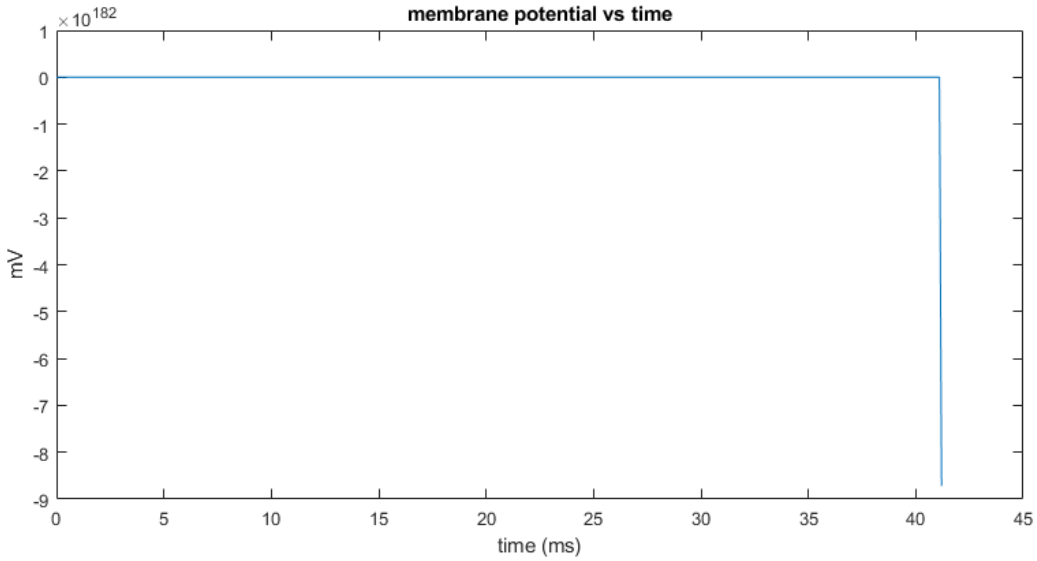
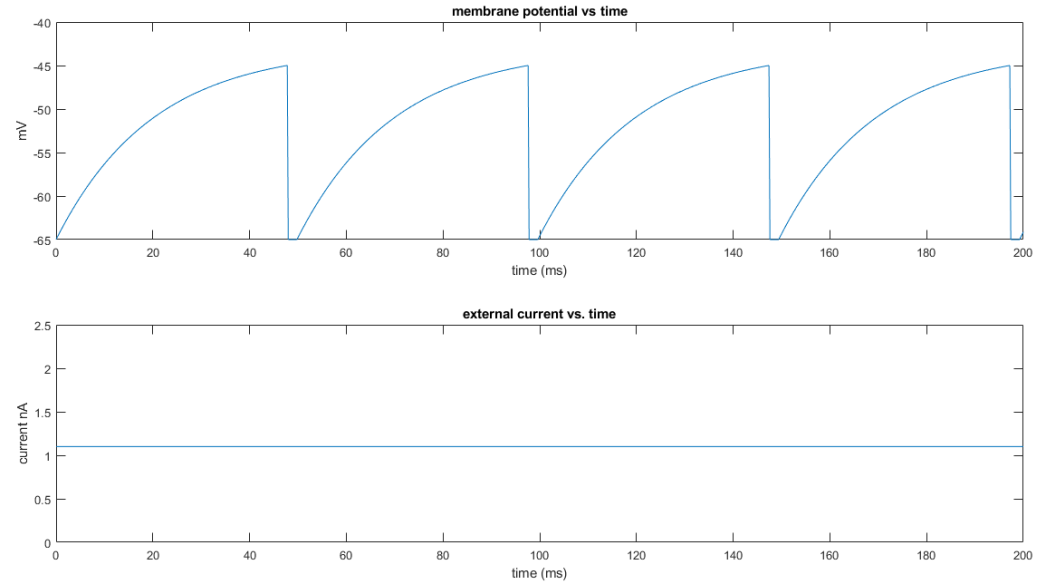


Figure PartB3i Graph of the HH model with a high dt will result in a poor simulation due large step size. The step size for this graph is 0.1 ms.



PartB3ii Graph of the LIF model with a dt of 0.2 will results in a graph that is like the model with a dt of 0.1.

Time (ms)	HH model	LIF model
0.1	0.012093	0.002786
0.01	0.052430	0.009561
0.002	0.196750	0.020118

Table PartB3iii the simulation time of the HH model and the LIF model at different step sizes. Since the HH model is more calculation intensive, it is not a surprise that it has a longer average simulation length compared to the LIF model. Regardless, even with a small step size of 0.002 the HH simulation can still be ran at a reasonable speed and there is no significant difference that would warrant choosing one over the other due to computation times.

Appendix

PartA1-5

```
%Constants
G_Na = 400;
G_K = 200;
G_L = 2;
E_Na = 99;
E_K = -85;
V_L = -65;
C = 2;

%Simulation parameters
t_max = 200;
dt = 0.01;
t_start = 40;

time = 0:dt:t_max;

I_0 = 200;
I_e = 0:dt:t_max;
for i = 1:t_max/dt
    if i < 40/dt
        I_e(i) = 0;
    else
        I_e(i) = I_0;
    end
end

%Spike detection
spikeRecorded = false;
totalSpikes = 0;
detecThreshold = -15;

%initial conditions
V(1) = V_L;
m(1) = a_m(V(1)) / (a_m(V(1)) + b_m(V(1)));
h(1) = a_h(V(1)) / (a_h(V(1)) + b_h(V(1)));
n(1) = a_n(V(1)) / (a_n(V(1)) + b_n(V(1)));
for i = 1:(t_max/dt)
    dVt1 = -G_L * (V(i) - V_L);
    dVt2 = -G_Na * power( m(i), 3) * h(i) * (V(i) - E_Na);
    dVt3 = -G_K * power( n(i), 4) * (V(i) - E_K);
    dVdt = (dVt1 +dVt2 + dVt3 + I_e(i)) / C;
    dmdt = a_m(V(i)) * (1 - m(i)) - b_m(V(i)) * m(i);
```

```
    dhdt = a_h(V(i)) * (1 - h(i)) - b_h(V(i)) * h(i);
    dndt = a_n(V(i)) * (1 - n(i)) - b_n(V(i)) * n(i);

    V(i + 1) = V(i) + dVdt * dt;
    m(i + 1) = m(i) + dmdt * dt;
    h(i + 1) = h(i) + dhdt * dt;
    n(i + 1) = n(i) + dndt * dt;

    I_m(i + 1) = -dVt1 - dVt2 - dVt3;               %total membrane current
    Na_c(i + 1) = G_Na * power( m(i), 3) * h(i);    %sodium conductance
    K_c(i + 1) = G_K * power( n(i), 4);             %potassium conductance

    if V(i + 1) > detecThreshold
        if spikeRecorded == false;
            totalSpikes = totalSpikes + 1;
            spikeRecorded = true;
        end
    else
        spikeRecorded = false;
    end
end
%Command Window Output
disp("total spikes: " + totalSpikes);

%Cleanup - A bit difficult to collect the initial data for the total
%membrane current, the sodium conductance, and the sodium conductance
I_m(1) = I_m(2);
Na_c(1) = Na_c(2);
K_c(1) = K_c(2);

%Figure plotting
figure(1);
subplot(5,1,1);
plot(time, V);
title('membrane potential vs time');
ylabel('mV');
xlabel('time (ms)');
subplot(5,1,2);
plot(time, I_m);
title('total membrane current vs time');
ylabel('mV');
xlabel('time (ms)');
subplot(5,1,3);
plot(time, Na_c);
title('sodium conductance vs time');
ylabel('mV');
xlabel('time (ms)');
subplot(5,1,4);
plot(time, K_c);
title('potassium conductance vs time');
ylabel('mV');
xlabel('time (ms)');
subplot(5,1,5);
plot(time, I_e);
title('current vs time');
```

```
ylabel('pA');
xlabel('time (ms)');

%Common Functions
function out = a_m(V)
    num = 0.1 * (V + 40);
    den = 1 - power(exp(1), (-0.1 * (V + 40)));
    out = num / den;
end

function out = b_m(V)
    out = 4 * power(exp(1), (-0.0556 * (V + 65)));
end

function out = a_h(V)
    out = 0.07 * power(exp(1), (-0.05 * (V + 65)));
end

function out = b_h(V)
    den = 1 + power(exp(1), (-0.1 * (V + 35)));
    out = 1 / den;
end

function out = a_n(V)
    num = 0.01 * (V + 55);
    den = 1 - power( exp(1), (-0.1 * (V + 55)));
    out = num / den;
end

function out = b_n(V)
    out = 0.125 * power( exp(1), (-0.0125 * (V + 65)));
end
```

Part B

```
clear
%Constants
G_L = 0.05;
V_L = -65;

V_spk = -45;
V_r = -65;
tau_arp = 2;

fireRate = [];
fireRateIndex = 1;
currentValues = 0:2:80;
%Simulation Settings
dt = 0.1;
t_max = 200;

%Initial Conditions
tic
for c = 0:2:80
    dtSpike = [];
```

```
    spikeIndex = 1;

    V(1) = V_L;
    I_e(1) = c;

    dt_tau = 0;
    for t = 1:t_max / dt

        dVdt = (-G_L * (V(t) - V_L)) + I_e(t);

        if t < dt_tau
            V(t + 1) = V(t);
            I_e(t + 1) = I_e(t);
            continue;
        end

        if V(t) > V_spk
            V(t + 1) = V_r;
            dt_tau = t + (tau_arp / dt);

            dtSpike(spikeIndex) = (t * dt) * (1/1000);
            spikeIndex = spikeIndex + 1;
        else
            V(t + 1) = V(t) + (dVdt * dt);
        end
        I_e(t + 1) = I_e(t);
    end

    sum = 0;
    for i = 1:(length(dtSpike) - 1)
        sum = sum + (dtSpike(i + 1) - dtSpike(i));
    end

    fireRate(fireRateIndex) = 1 / (sum / length(dtSpike));
    fireRateIndex = fireRateIndex + 1;
end
toc

plot(currentValues, fireRate);
title('Firerate vs Current');
xlabel('current (nA)');
ylabel('FireRate');
```