Excercise 3

Contents

question

Part 1

- I. Plot the voltage-dependent functions n, m and h in steady state as a function of V .
- II. Plot the voltage-dependent time constants of n, m and h as a function of V .

```
clc
close all
clear

total_data_points = 2000;
dt = 0.1;             %ms
t = (0:total_data_points) * dt;
cm = 10;             %nF/mm^2
V_init = -65;      %mv
m_init = 0.0529;
h_init = 0.5961;
n_init = 0.3177;
g_L = 0.3;          %mS/mm^2
g_K = 36;           %mS/mm^2
g_Na = 120;       %mS/mm^2
E_L = -54.387;  %mv
E_K = -77;         %mv
E_Na = 50;        %mv

V = linspace(-100, 100, total_data_points);

n(1) = n_init;
m(1) = m_init;
h(1) = h_init;

for i = 1:total_data_points
      alpha_n(i) = 0.01*(V(i) + 55) / (1 - exp(-0.1*(V(i) + 55)));    beta_n(i) = 0.125*exp(-0.0125*(V(i) + 65));
      alpha_m(i) = 0.1*(V(i) + 40) / (1 - exp(-0.1*(V(i) + 40)));     beta_m(i) = 4*exp(-0.0556*(V(i) + 65));
      alpha_h(i) = 0.07*exp(-0.05*(V(i) + 65));                           beta_h(i) = 1/(1 + exp(-0.1*(V(i) + 35)));

      n_inf(i) = alpha_n(i)   / (alpha_n(i) + beta_n(i));
      m_inf(i) = alpha_m(i) / (alpha_m(i) + beta_m(i));
      h_inf(i) = alpha_h(i)  / (alpha_h(i) + beta_h(i));

     Tau_n(i)  = 1 / (alpha_n(i) + beta_n(i));
     Tau_m(i) = 1 / (alpha_m(i) + beta_m(i));
     Tau_h(i)  = 1 / (alpha_h(i) + beta_h(i));

end
```

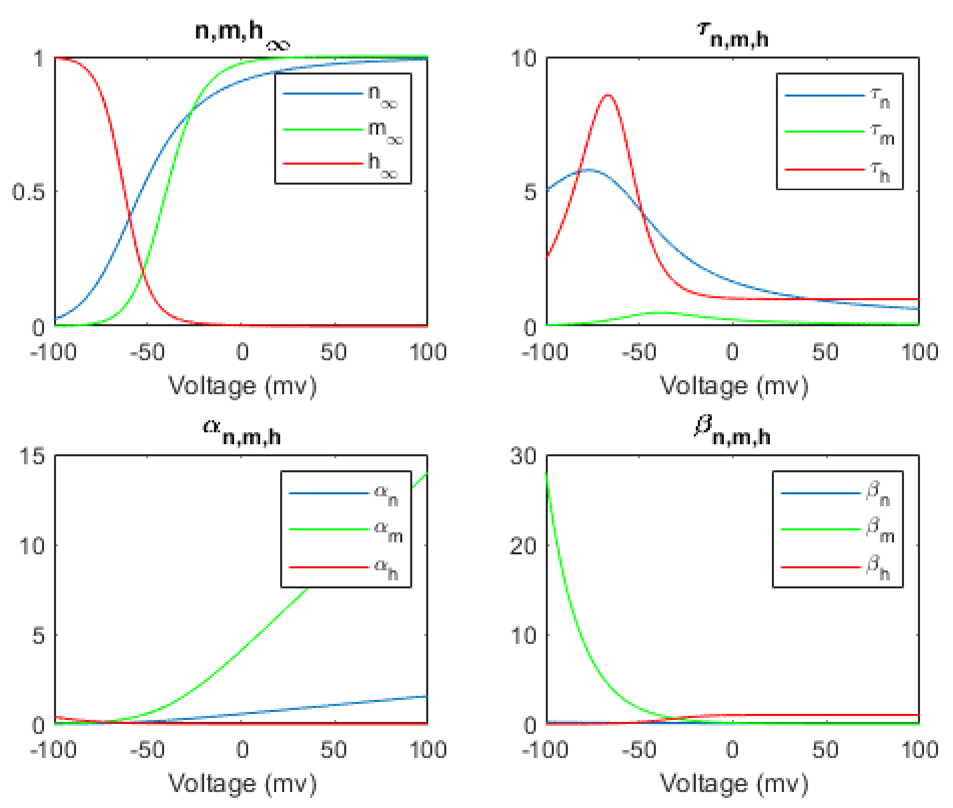
let's plot alpha_n,m,h & n,m,h _ inf & Taw n,m,h

```
figure
subplot(2,2,1)
plot(V , n_inf(:)),     hold on,             plot(V, m_inf (:), 'g'),     hold on,     plot(V, h_inf (:), 'r' ),
title('n,m,h_\infty'),       xlabel('Voltage (mv)'),
legend('n_\infty', 'm_\infty', 'h_\infty');

subplot(2,2,2)
plot(V , Tau_n(:)),     hold on,           plot(V, Tau_m(:), 'g'),      hold on,     plot(V, Tau_h(:) , 'r' )
title('\tau_n_,_m_,_h')
xlabel('Voltage (mv)')
legend('\tau_n', '\tau_m', '\tau_h');

subplot(2,2,3)
plot(V , alpha_n(:)),     hold on,           plot(V, alpha_m(:), 'g'),      hold on,     plot(V, alpha_h(:) , 'r' )
title('\alpha_n_,_m_,_h')
xlabel('Voltage (mv)')
legend('\alpha_n', '\alpha_m', '\alpha_h');

subplot(2,2,4)
plot(V , beta_n(:)),     hold on,           plot(V, beta_m(:), 'g'),      hold on,     plot(V, beta_h(:) , 'r' )
title('\beta_n_,_m_,_h')
xlabel('Voltage (mv)')
legend('\beta_n', '\beta_m', '\beta_h');
```

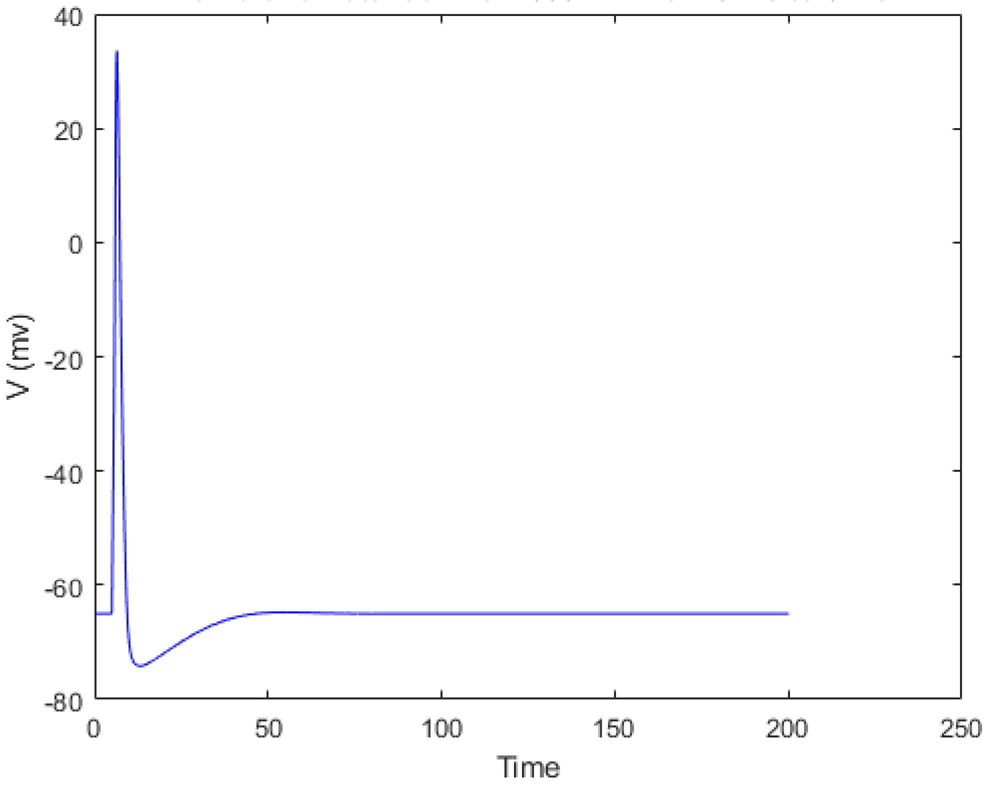


part 2

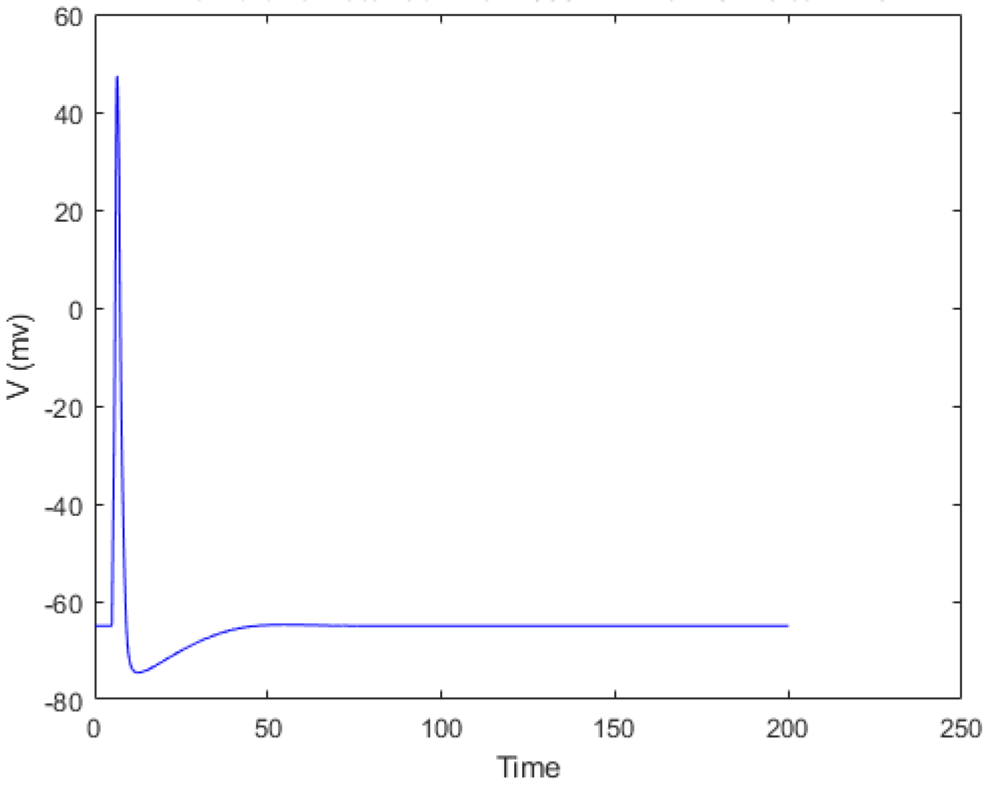
2)Plot voltage V as function of time , by simulating the injection of a suitable short (try some values!) external current Ie ,starting at t = 5ms.

```
close all
Action_Potential_fun(120, 5, 6, 600)
Action_Potential_fun(120, 5, 7, 600)
Action_Potential_fun(120, 5, 9, 600)
Action_Potential_fun(120, 5, 11, 600)
Action_Potential_fun(120, 5, 14, 600)
Action_Potential_fun(120, 5, 18, 600)
```

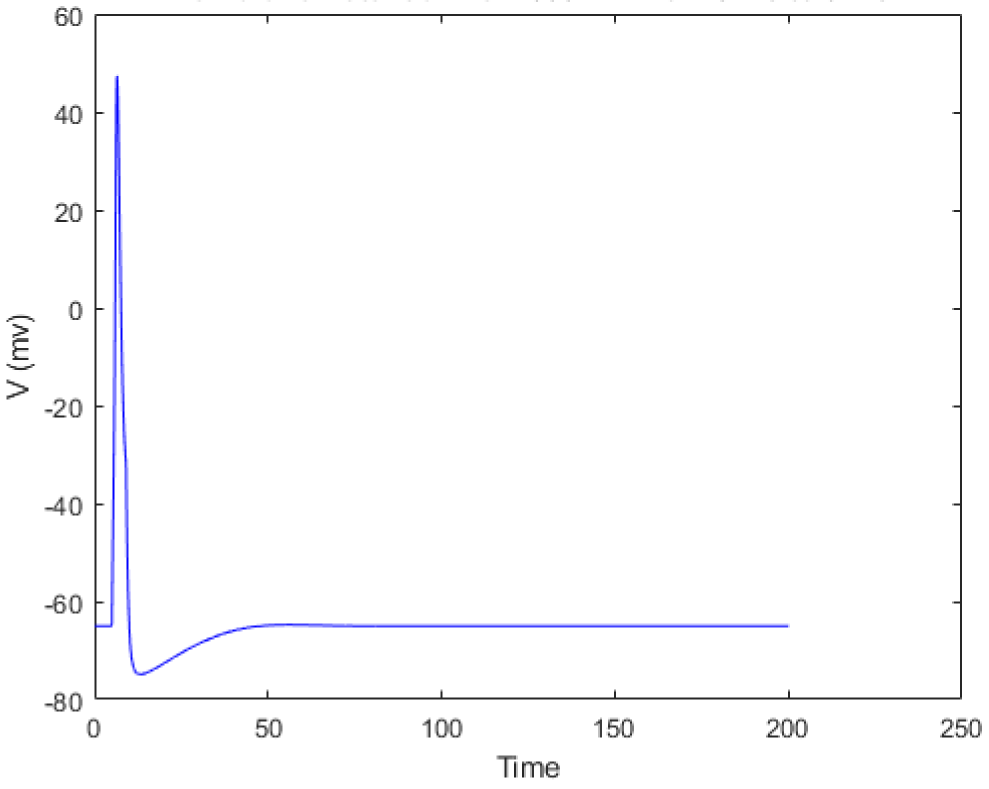
Membrane Potential : Ie = 600 mA from 5 ms to 6 ms



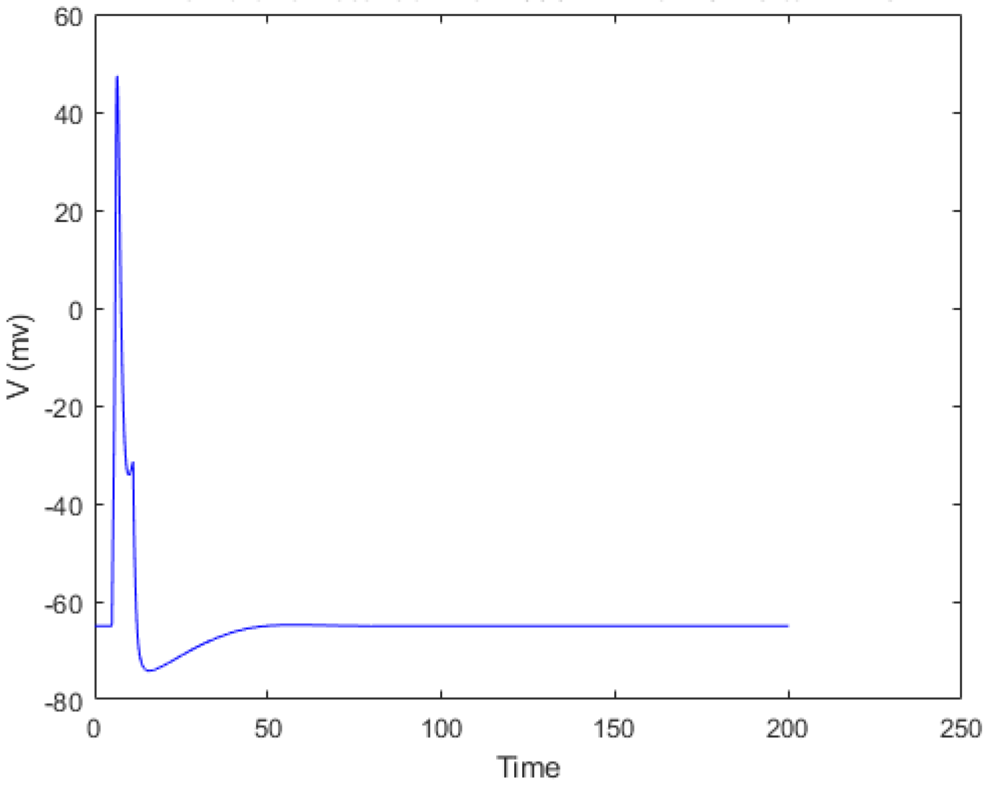
Membrane Potential : Ie = 600 mA from 5 ms to 7 ms



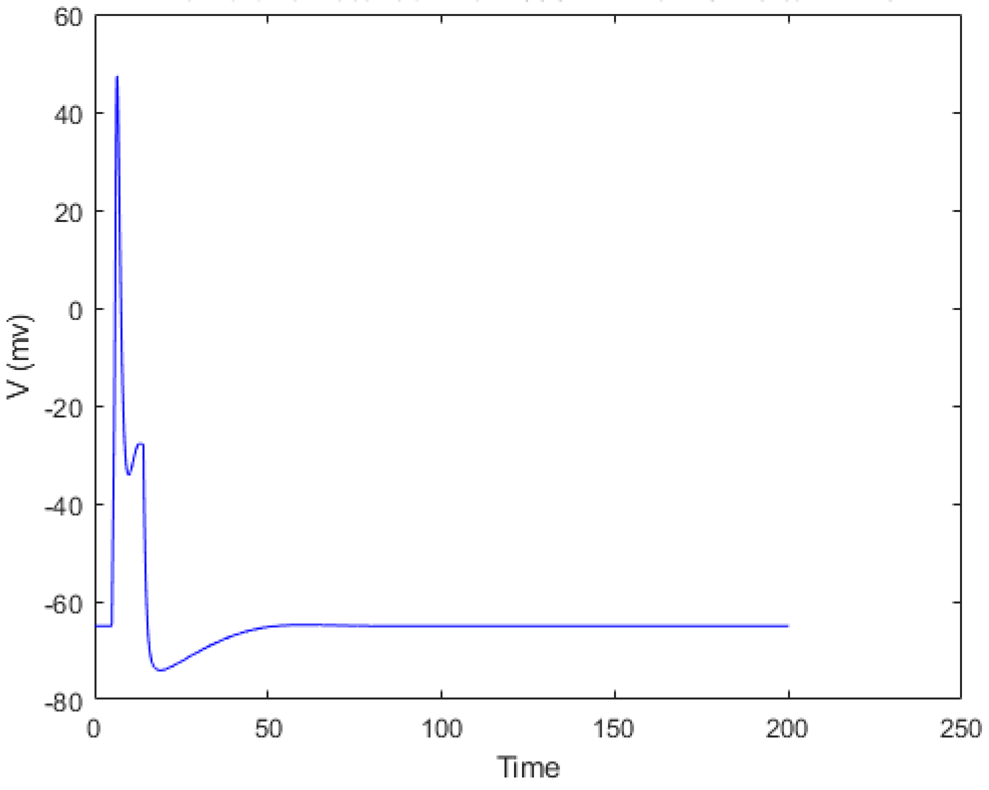
Membrane Potential : Ie = 600 mA from 5 ms to 9 ms



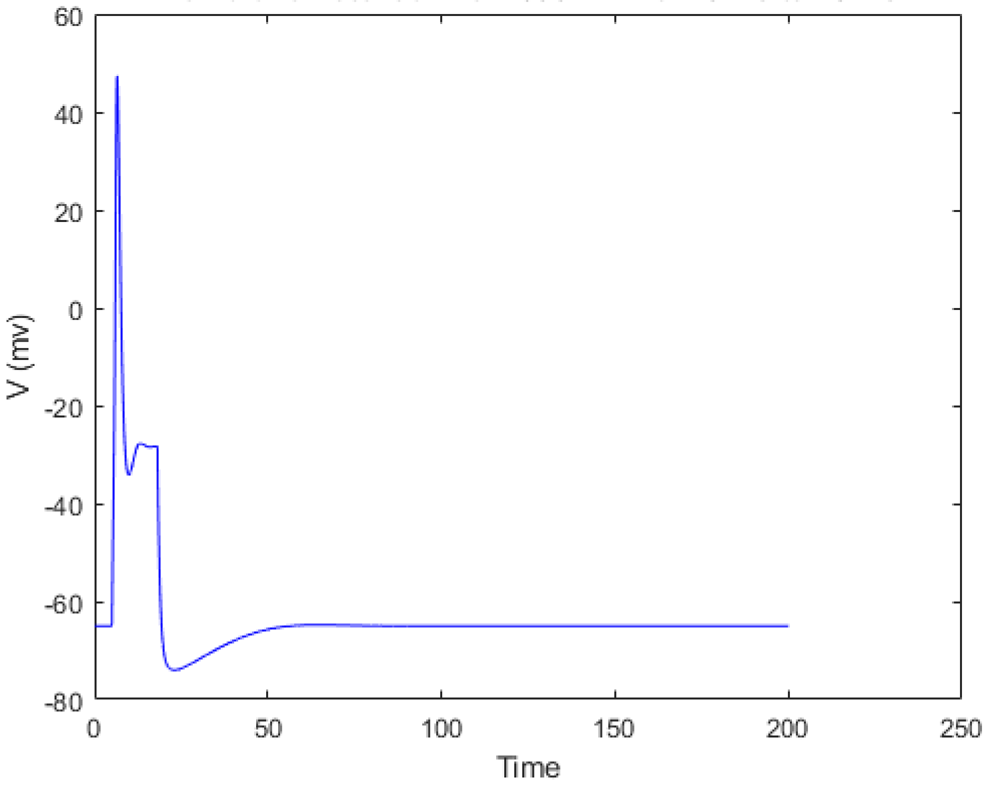
Membrane Potential : Ie = 600 mA from 5 ms to 11 ms



Membrane Potential : Ie = 600 mA from 5 ms to 14 ms



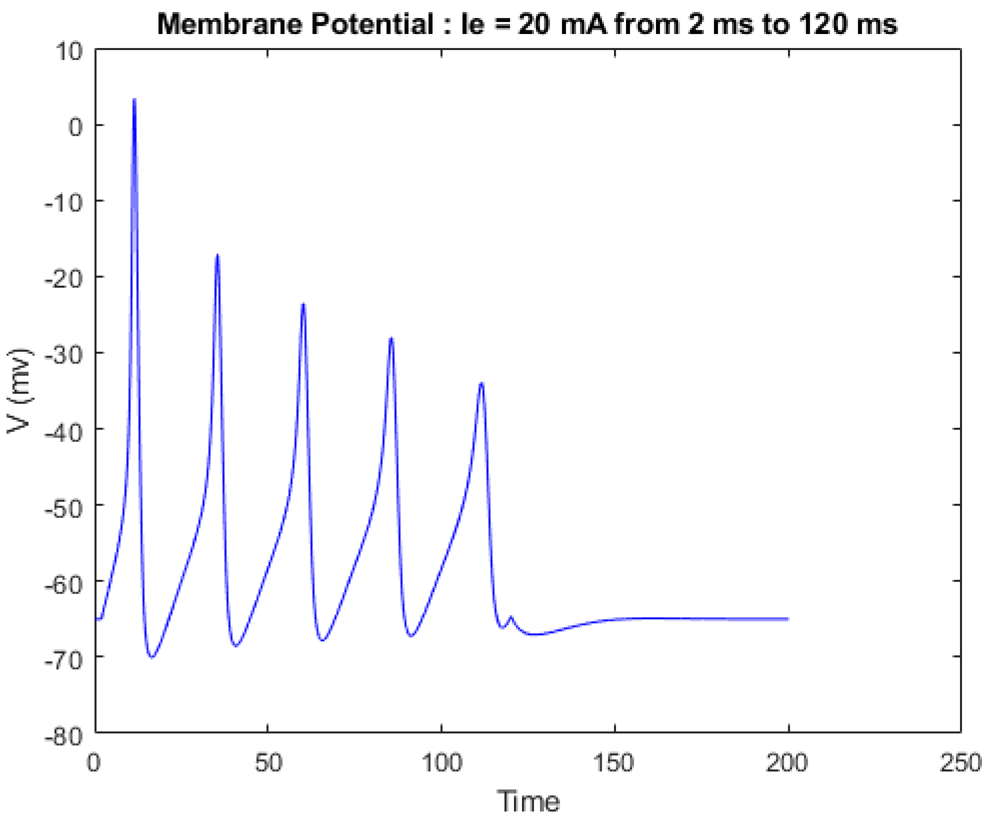
Membrane Potential : Ie = 600 mA from 5 ms to 18 ms

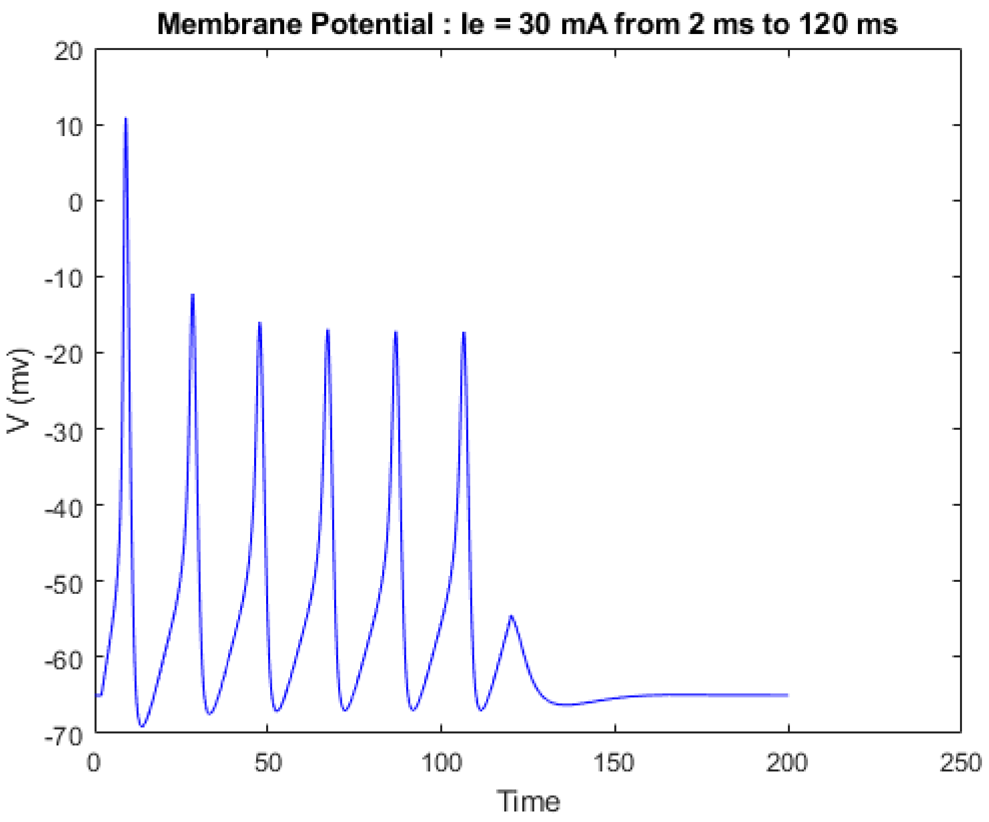


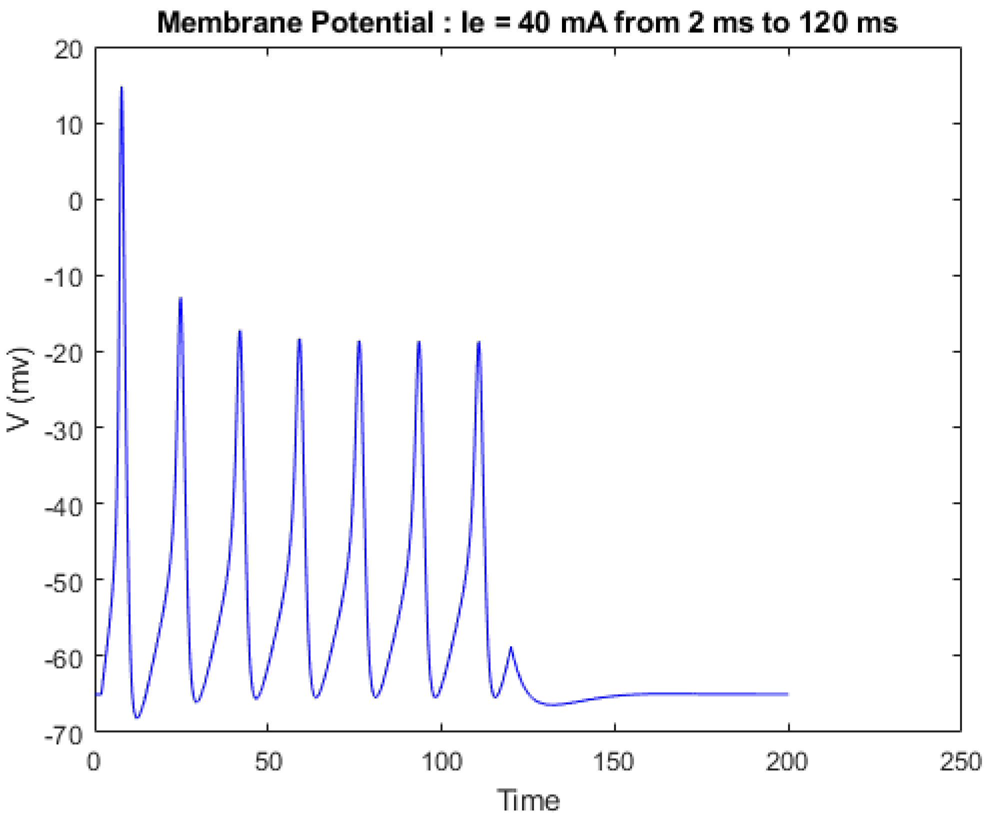
part 3

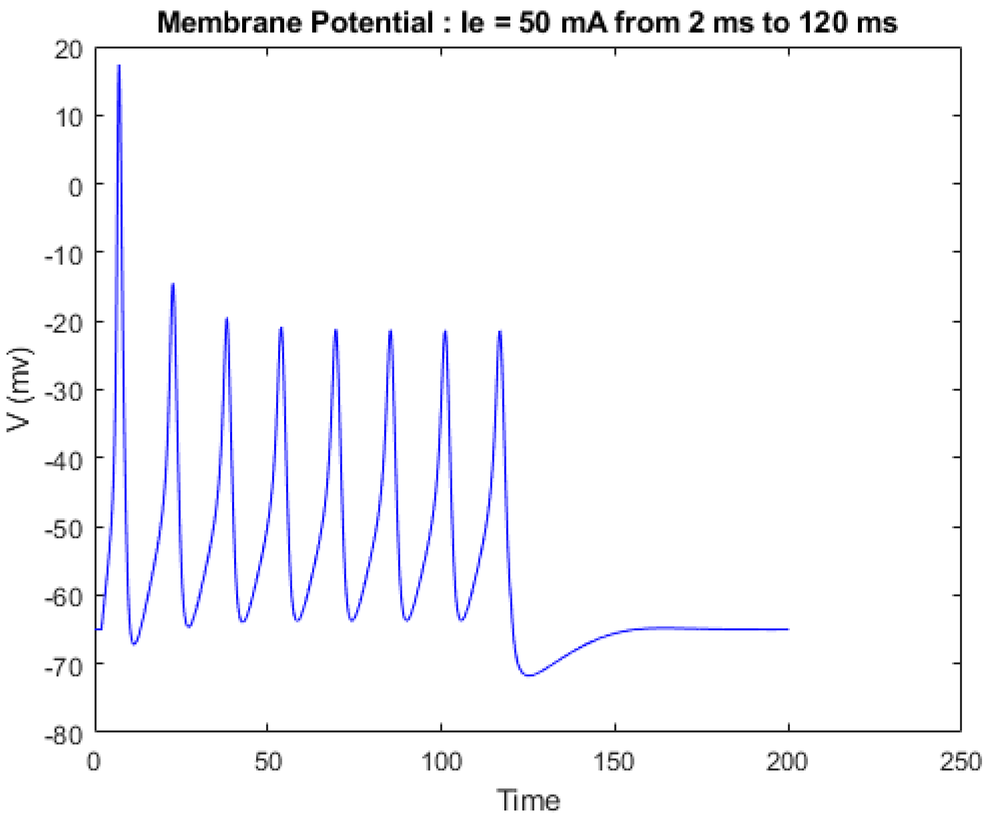
3)Apply constant input from 2 msto 120 msand see the result. Increase the sodium conductance 5 times and then see the result.

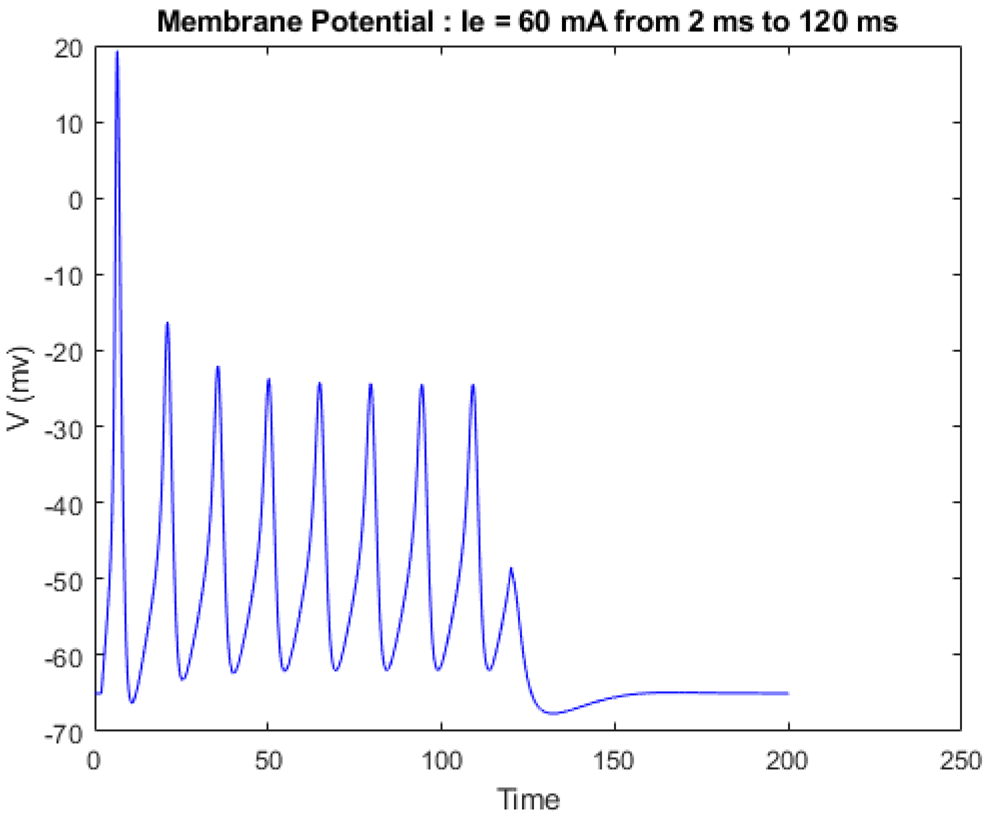
```
close all
Action_Potential_fun(120, 2, 120, 20)
Action_Potential_fun(120, 2, 120, 30)
Action_Potential_fun(120, 2, 120, 40)
Action_Potential_fun(120, 2, 120, 50)
Action_Potential_fun(120, 2, 120, 60)
Action_Potential_fun(120, 2, 120, 200)
```

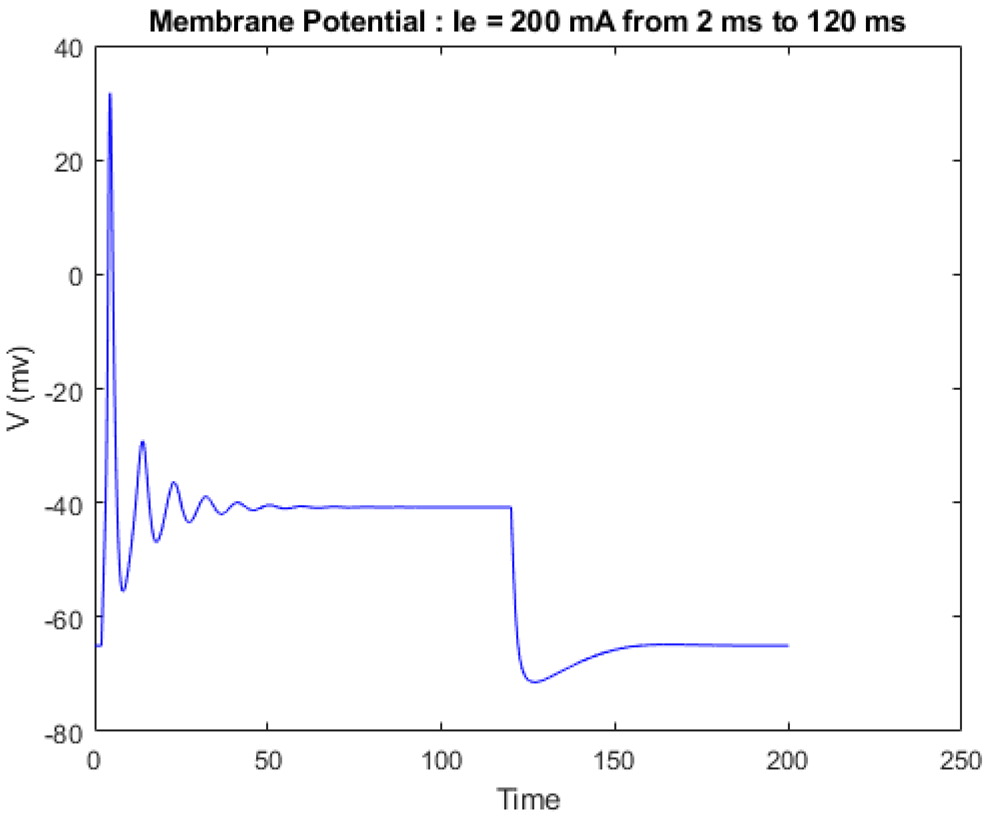

Membrane Potential : Ie = 20 mA from 2 ms to 120 ms


Membrane Potential : Ie = 30 mA from 2 ms to 120 ms


Membrane Potential : Ie = 40 mA from 2 ms to 120 ms

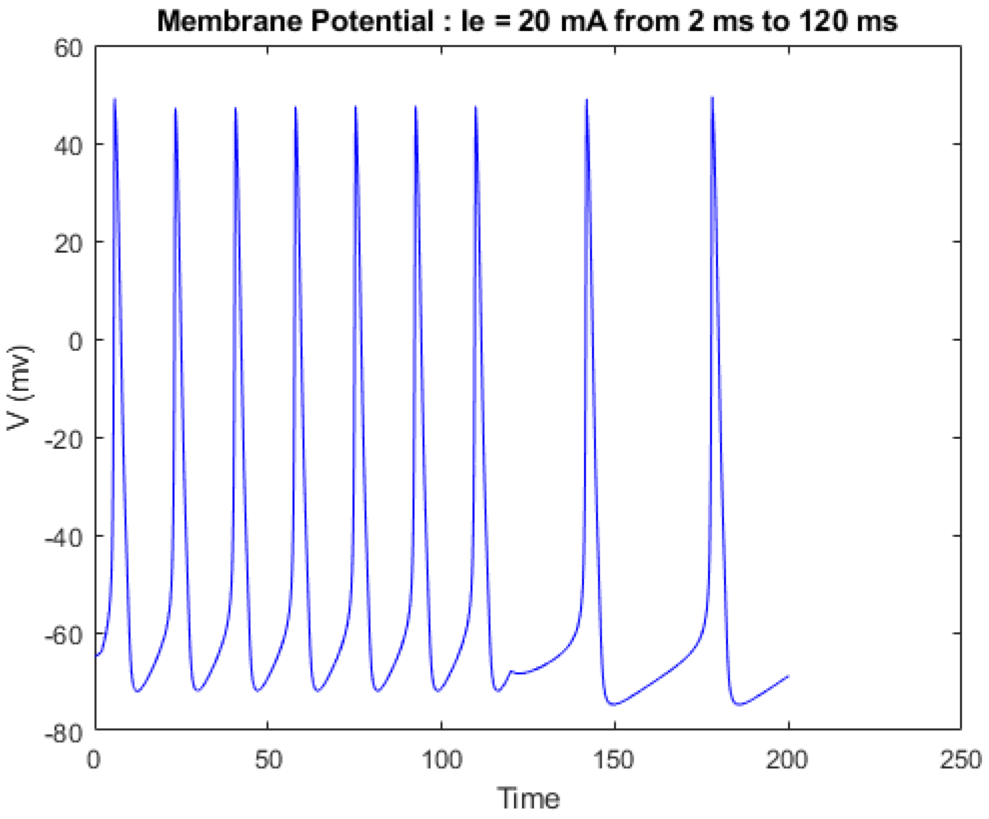

Membrane Potential : Ie = 50 mA from 2 ms to 120 ms


Membrane Potential : Ie = 60 mA from 2 ms to 120 ms

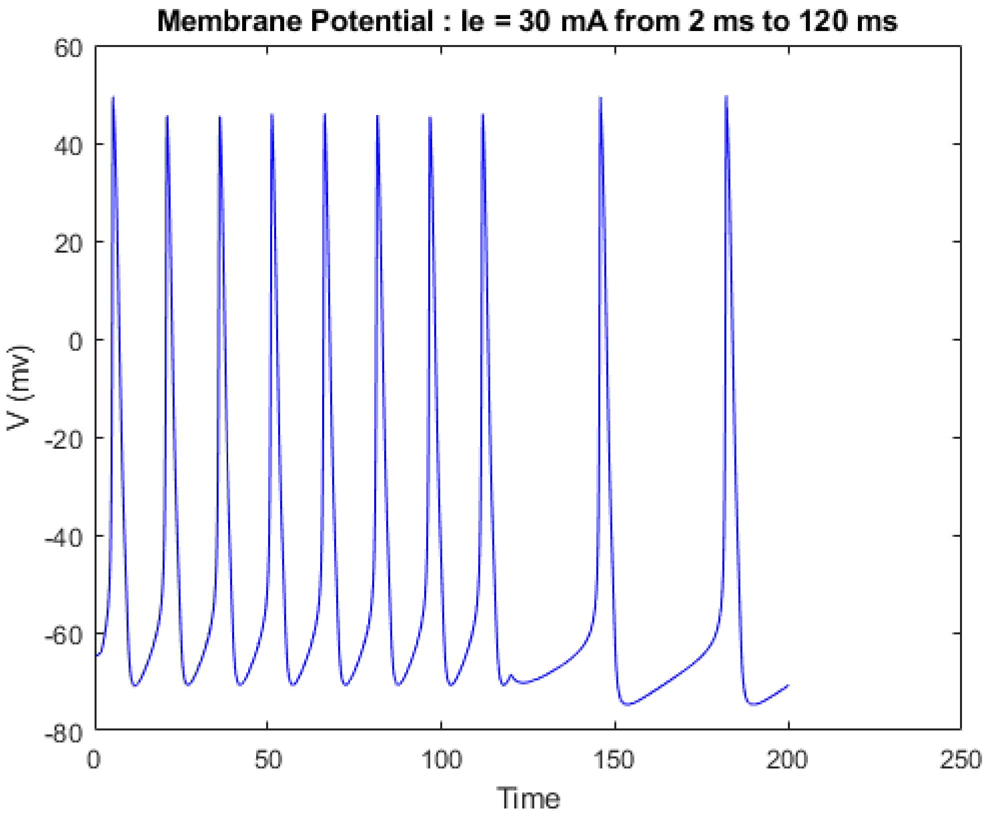

Membrane Potential : Ie = 200 mA from 2 ms to 120 ms

Now Let's multiply the Na conductance by 5

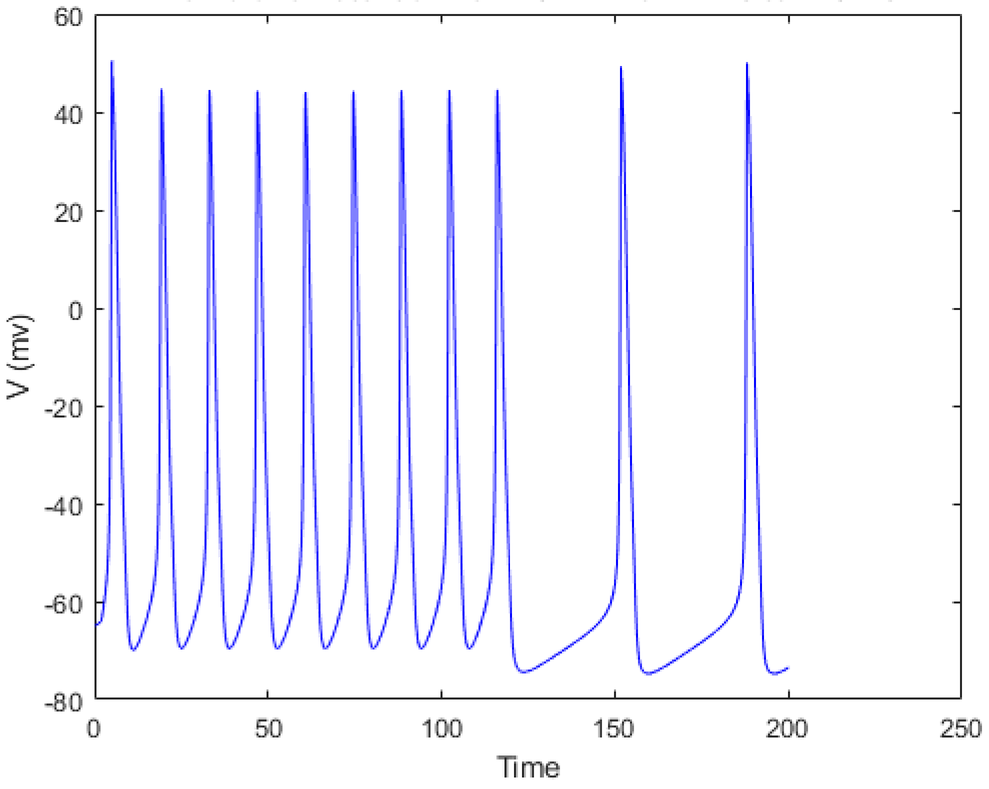
```
close all
Action_Potential_fun(600, 2, 120, 20)
Action_Potential_fun(600, 2, 120, 30)
Action_Potential_fun(600, 2, 120, 40)
Action_Potential_fun(600, 2, 120, 70)
Action_Potential_fun(600, 2, 120, 200)
Action_Potential_fun(600, 2, 120, 600)
```



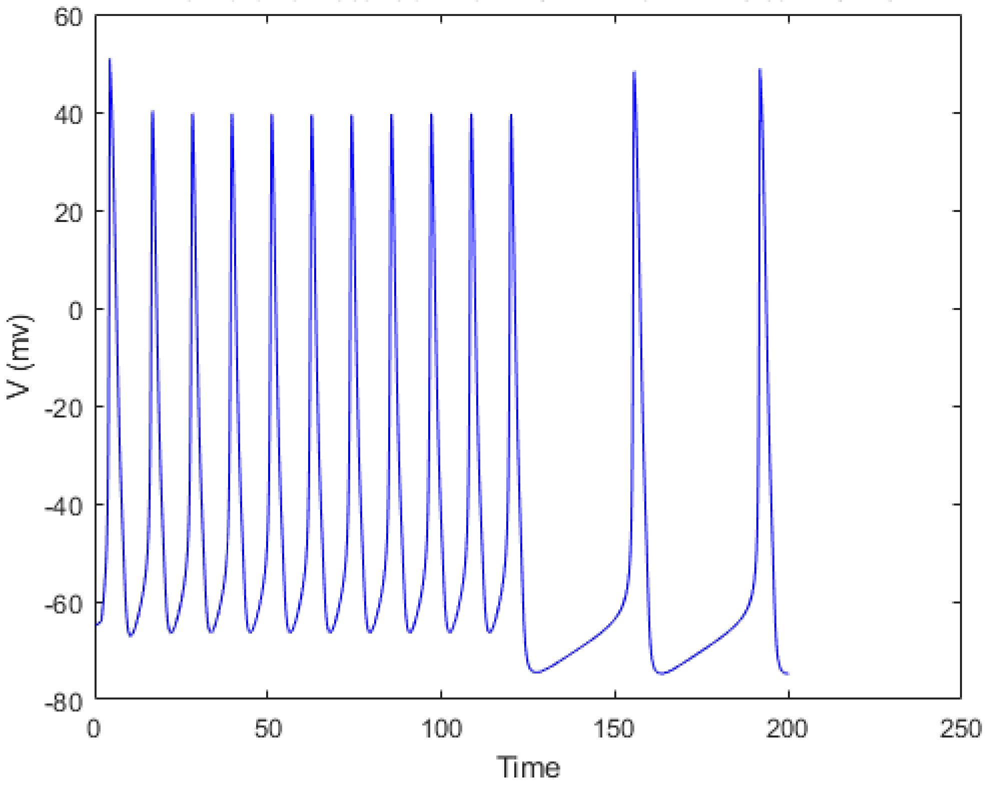
Membrane Potential : Ie = 20 mA from 2 ms to 120 ms



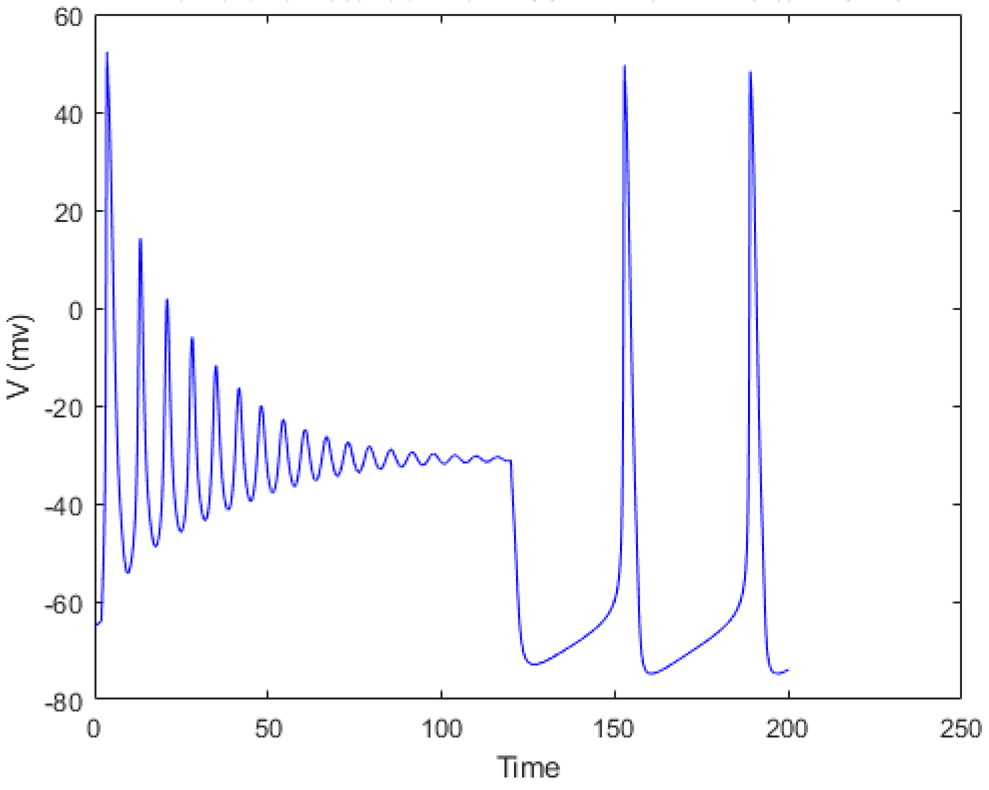
Membrane Potential : Ie = 30 mA from 2 ms to 120 ms



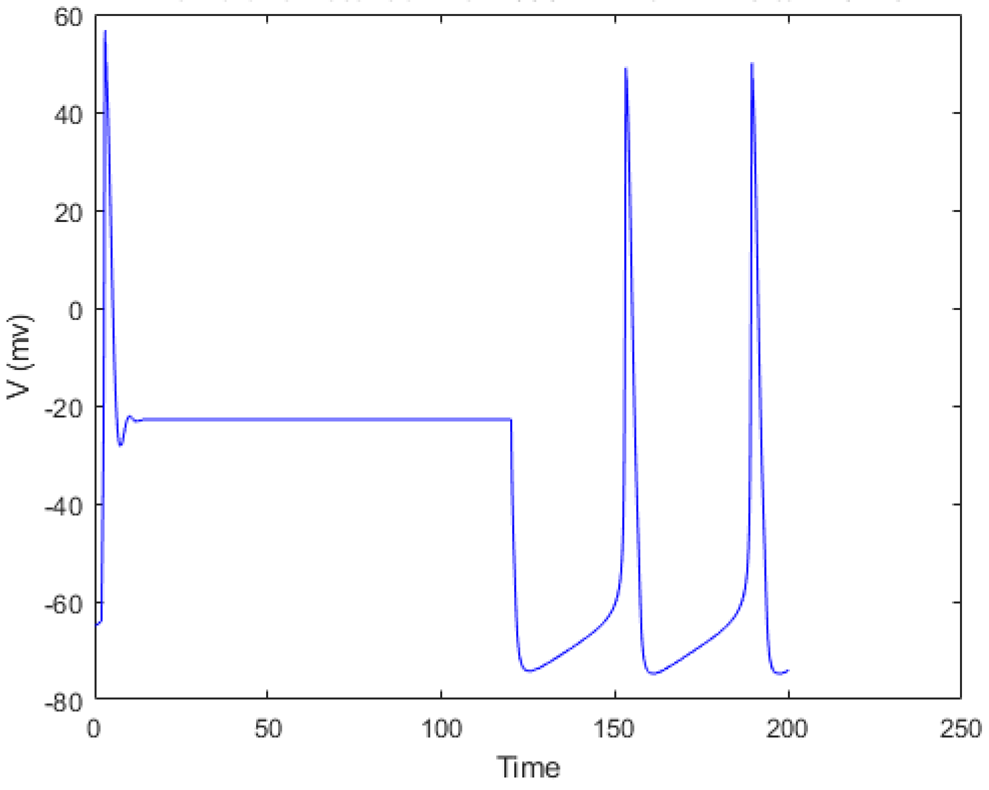
Membrane Potential : Ie = 40 mA from 2 ms to 120 ms



Membrane Potential : Ie = 70 mA from 2 ms to 120 ms



Membrane Potential : Ie = 200 mA from 2 ms to 120 ms



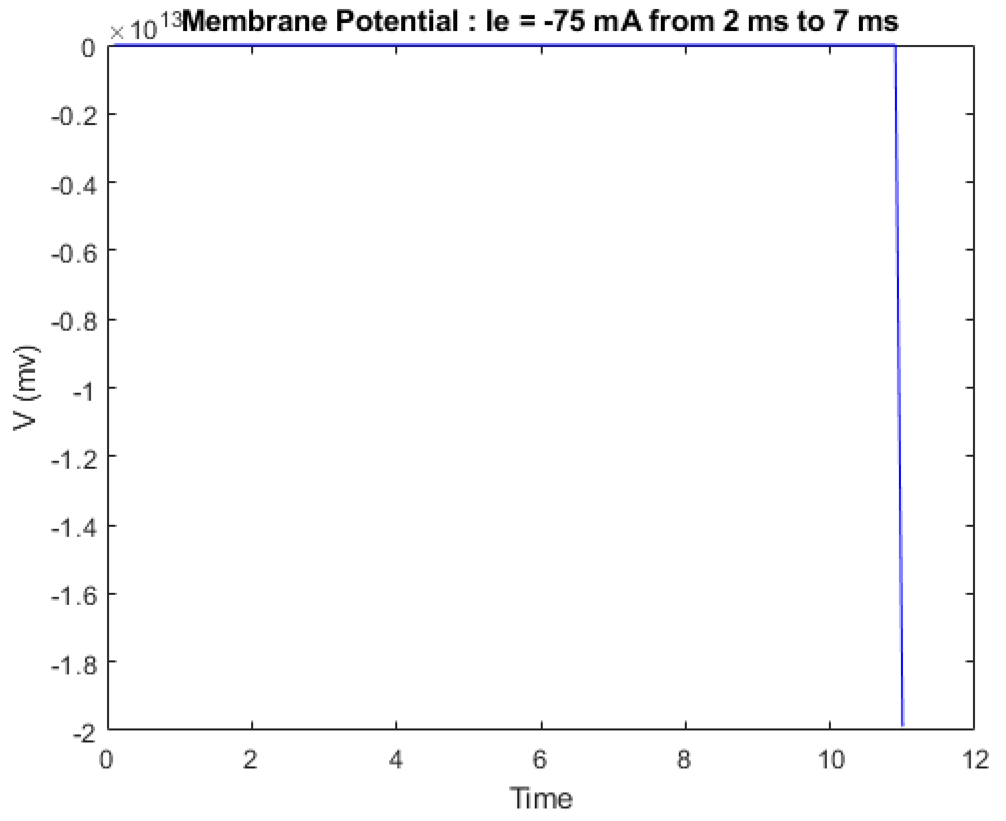
Membrane Potential : Ie = 600 mA from 2 ms to 120 ms

part 4

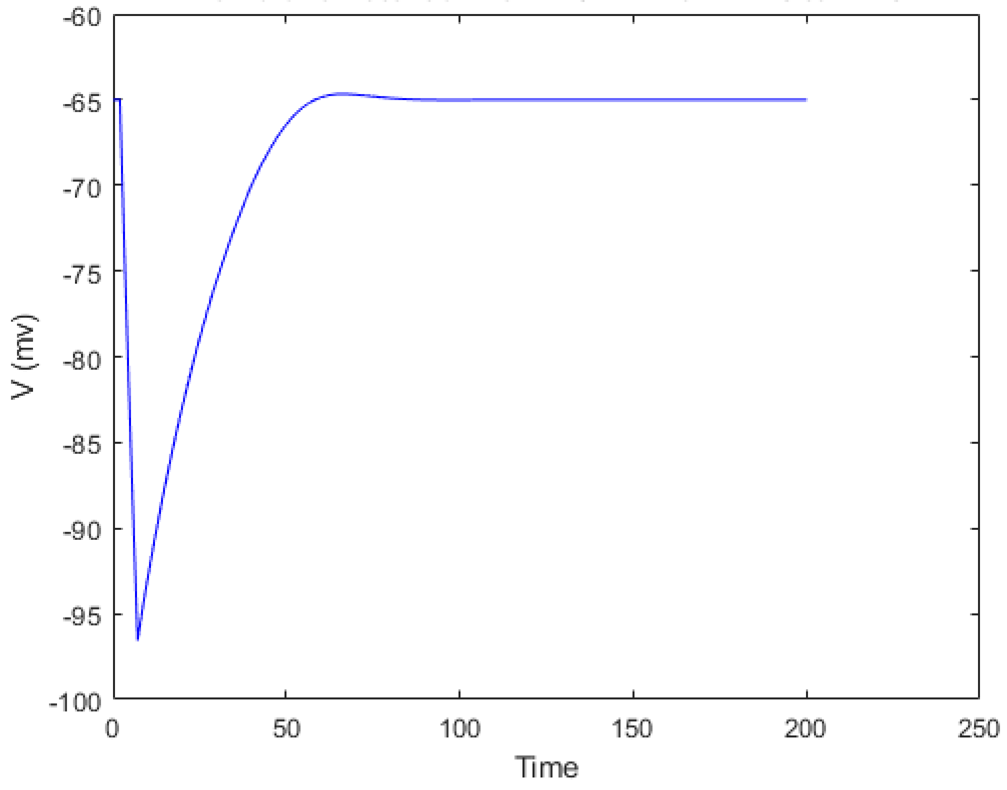
4)Apply inhibitory input from 2 ms to 7 ms and then set the input current to 0. What happens?

```
close all
Action_Potential_fun(120, 2, 7, -75)
Action_Potential_fun(120, 2, 7, -70)
Action_Potential_fun(120, 2, 7, -50)
Action_Potential_fun(120, 2, 7, -30)
Action_Potential_fun(120, 2, 7, -10)
Action_Potential_fun(120, 2, 7, 0)
```

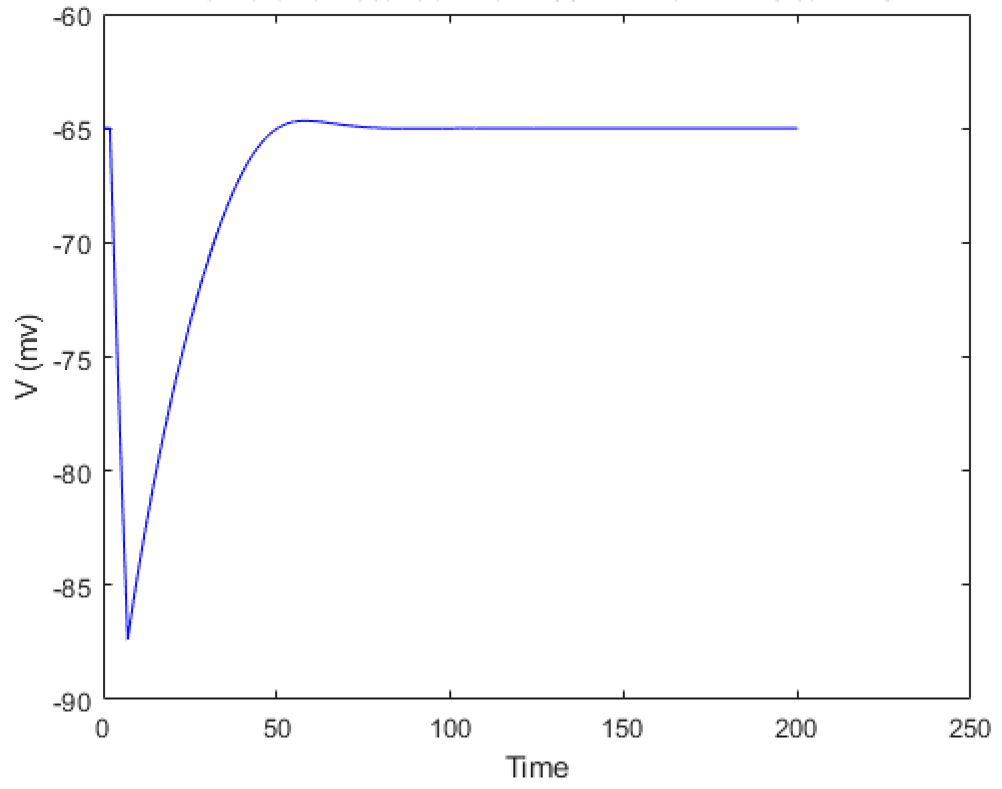
Membrane Potential : Ie = -75 mA from 2 ms to 7 ms



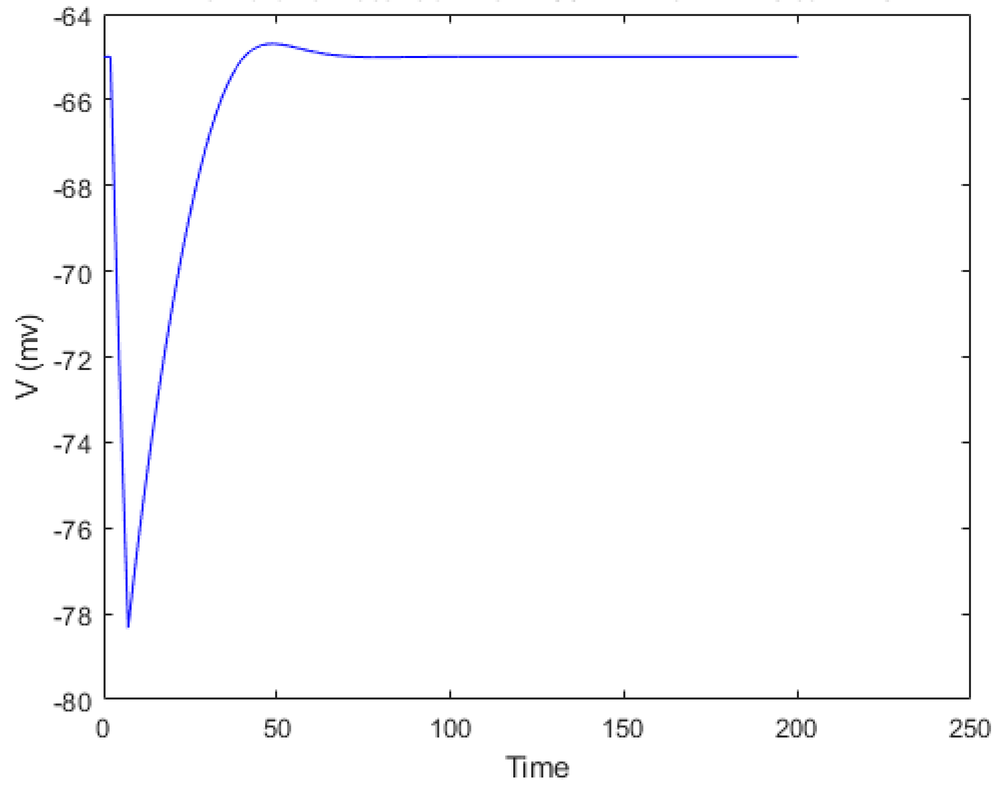
Membrane Potential : Ie = -70 mA from 2 ms to 7 ms



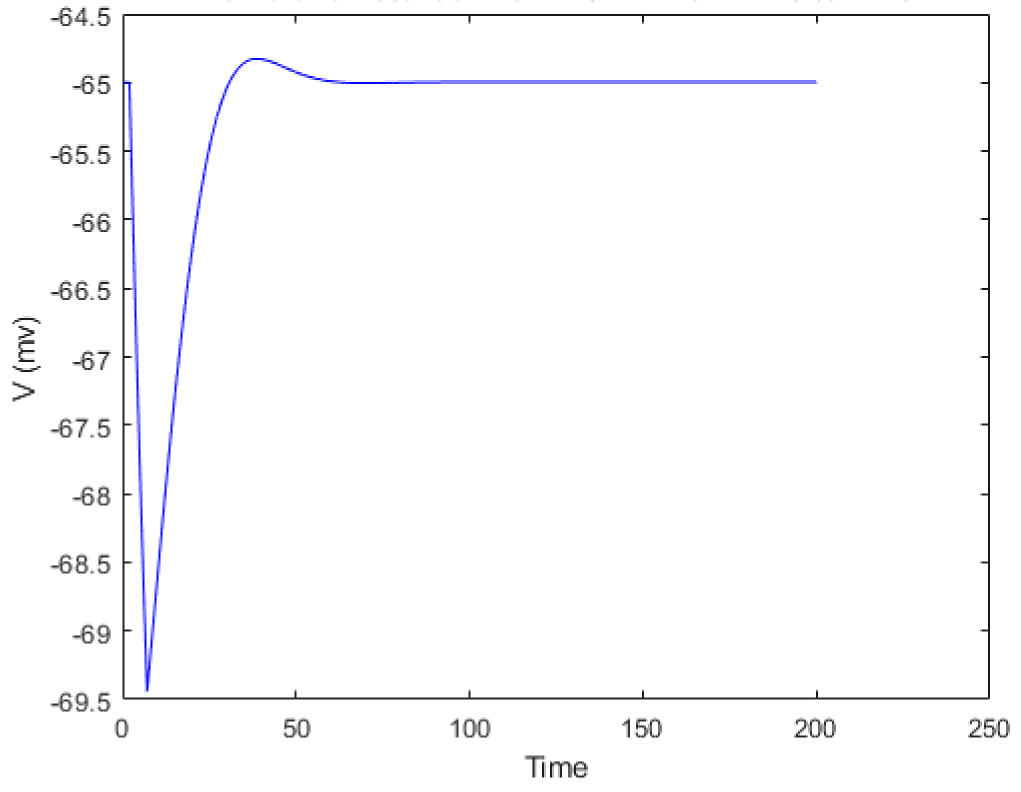
Membrane Potential : Ie = -50 mA from 2 ms to 7 ms



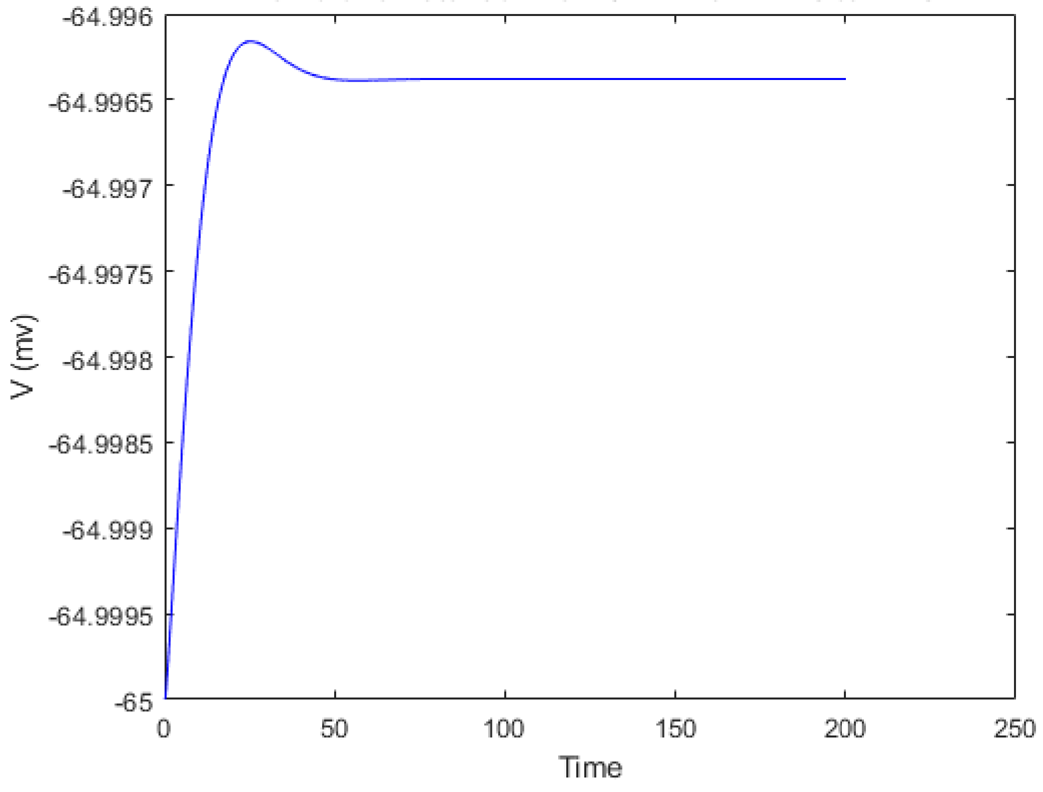
Membrane Potential : Ie = -30 mA from 2 ms to 7 ms



Membrane Potential : Ie = -10 mA from 2 ms to 7 ms



Membrane Potential : Ie = 0 mA from 2 ms to 7 ms



```
clc
```

Explanation of the plots:

Regarding the effect of duration and input amplitude value, the input value is important, in low values only depolarization occurs, but in stronger amplitude values, the neuron spikes.

Regarding the duration, if the input is applied for 120 milliseconds, after that the neuron enters the sustained periodic response phase and spikes regularly even though the stimulation is stopped.

These show that the duration of stimulation and the amount of stimulation amplitude both have an effect on the behavior of the neuron, and sometimes they have non-obvious effects: as in the fourth part, when an inhibitory stimulus is applied to a neuron, after the stimulation is removed, we might expect the neuron's voltage to change. but on the contrary, the neuron spikes regularly. This effect is called post inhibitory rebound.

This exercise demonstrates the complex and non-trivial behavior of the HH model.

```
function Action_Potential_fun(g_Na, start_time, end_time, Ie_val)
    total_data_points = 2000;
    dt = 0.1;             %ms
    cm = 10;              %nF/mm^2
    V_init = -65;      %mv
    m_init = 0.0529;
    h_init = 0.5961;
    n_init = 0.3177;
    g_L = 0.3;           %mS/mm^2
    g_K = 36;            %mS/mm
    E_L = -54.387;  %mv
    E_K = -77;          %mv
    E_Na = 50;         %mv
    A = 1;

    n(1) = n_init;
    m(1) = m_init;
    h(1) = h_init;

    V(1) = -65; %mv

    Ie = zeros (total_data_points ,1);
    Ie(start_time/dt: end_time/dt) = Ie_val;
    for i =1:total_data_points
            alpha_n(i) = 0.01*(V(i) + 55) / (1 - exp(-0.1*(V(i) + 55)));    beta_n(i) = 0.125*exp(-0.0125*(V(i) + 65));
            alpha_m(i) = 0.1*(V(i) + 40) / (1 - exp(-0.1*(V(i) + 40)));     beta_m(i) = 4*exp(-0.0556*(V(i) + 65));
            alpha_h(i) = 0.07*exp(-0.05*(V(i) + 65));                          beta_h(i) = 1/(1 + exp(-0.1*(V(i) + 35)));

            n_inf(i) = alpha_n(i)   / (alpha_n(i) + beta_n(i));
            m_inf(i) = alpha_m(i) / (alpha_m(i) + beta_m(i));
            h_inf(i) = alpha_h(i)  / (alpha_h(i) + beta_h(i));

            Tau_n(i)  = 1 / (alpha_n(i) + beta_n(i));
            Tau_m(i) = 1 / (alpha_m(i) + beta_m(i));
            Tau_h(i)  = 1 / (alpha_h(i) + beta_h(i));

            % cm dV/dt= -im + Ie/A
            % im=gl ( V - El) + gk n^4 (V-Ek) + gNa m^3 h (V-ENa)
            im (i+1) = g_Na * m(i)^4  * (V(i) - E_Na) + g_K * n(i)^4 * (V(i) - E_K) + g_L * (V(i) - E_L);
            V(i+1)= V(i)- dt/cm * (g_Na * m(i)^3 *  h(i) * (V(i) - E_Na) + g_K * n(i)^4 * (V(i) - E_K) + g_L * (V(i) - E_L))+ dt/cm * Ie(i)/A;
            n(i+1) = n(i)   + dt*(n_inf(i) - n(i))/Tau_n(i);
            m(i+1) = m(i) + dt*(m_inf(i) - m(i))/Tau_m(i);
            h(i+1) = h(i)  + dt*(h_inf(i) - h(i))/Tau_h(i);

    end

    % Let's plot the membrane potential vursus time
    figure
    time = dt * (1:total_data_points+1);
    plot(time , V,'b'),     title(sprintf("Membrane Potential : Ie = %d mA from %d ms to %d ms",Ie_val, start_time, end_time)),
    xlabel('Time'),         ylabel('V (mv)')
end
```

Published with MATLAB® R2023a