Tutorial 2-1:

```
close all
clear all
clc
% Define parameters
C = 2e-9;  % Membrane capacitance (F)
R = 5*1e6;     % Membrane resistance (?)
leak_potential = -70e-3;  % Leak potential (V)
threshold = -50e-3;      % Spike threshold (V)
reset_potential = -65e-3;  % Reset potential (V)
dt = 0.1e-3;     % Time step (s)
T = 2;      % Maximum time (s)

% Create time vector
t = 0:dt:T;

% I define the I_app and the loop after that in Iapp function
I_th = (threshold  - leak_potential)/R;
I_lower = I_th*0.99;
[I_app_lower,v_lower] = Iapp(I_lower,
t,leak_potential,C,R,dt,threshold,reset_potential);
I_higher = I_th*1.01;
[I_app_higher,v_higher] = Iapp(I_higher,
t,leak_potential,C,R,dt,threshold,reset_potential);

img=figure;
plot(t(1:2001), v_lower(1:2001), 'b', 'LineWidth', 2);
hold on;
plot(t(1:2001), v_higher(1:2001), 'r', 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Membrane Potential (V)');
title('Membrane Potential with Different Applied Currents');
legend('slightly lower than','slightly higher than','Location','SouthEast');
grid on;
saveas(img, sprintf('Tutorial_2_1_question_1b.png'));
%Lower Current: I_low doesn't produce spikes, the calculation and simulation
confirm the minimum current threshold for spiking.
%Higher Current:I_high produces spikes, it verifies that currents above the
threshold trigger spiking behavior.
Tau = C*R;
i_0 = [I_th*1.05 I_th*1.1 I_th*1.15 I_th*1.2 I_th*1.25 I_th*1.30 I_th*1.25
I_th*1.35 I_th*1.4 I_th*1.45];
frates = zeros(size(i_0));
k=1;
for I_0 = i_0

    [I_app_f,v_f]  = Iapp(I_0, t, leak_potential, C, R, dt, threshold,
reset_potential);

    spike_count = 0;

    for n = 1:length(v_f)
        if v_f(n) == reset_potential;
            spike_count = spike_count + 1;
        end
```

```
    end
    rate = spike_count/2;
    frates(k) = rate;
    k=k+1;
end
img1=figure
scatter(i_0, frates);
xlabel('I_app');
ylabel('Firing rate(Hz)');
title('Firing rate based on injected current 1c');
legend('1c','Location','SouthEast');
grid on;
saveas(img1, sprintf('Tutorial_2_1_question_1c.png'));

f=zeros(size(i_0));
j=1;
for I_0 = i_0
    if ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - reset_potential) > 0) & ...
       ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - threshold) > 0)

        firing_rate = 1 / (Tau * log((leak_potential + R * Iapp(I_0, t,
leak_potential, C, R, dt, threshold, reset_potential) - reset_potential) /
...
                        (leak_potential + R * Iapp(I_0, t, leak_potential, C,
R, dt, threshold, reset_potential) - threshold)));

    end

    f(j) = firing_rate;
    j = j + 1;
end

img2=figure
scatter(i_0, frates);
hold on
scatter(i_0, f);
xlabel('I_app');
ylabel('Firing rate(Hz)');
title('Firing rate based on injected current');
legend('1c','1d','Location','SouthEast');
grid on;
saveas(img2, sprintf('Tutorial_2_1_question_1d.png'));
sigma=[0.1,0.2]

I_lower = I_th*0.99;
[I_app_lower_n,v_lower_n] = n_Iapp(I_lower,
t,leak_potential,C,R,dt,threshold,reset_potential,sigma(1));
I_higher = I_th*1.01;
[I_app_higher_n,v_higher_n] = n_Iapp(I_higher,
t,leak_potential,C,R,dt,threshold,reset_potential,sigma(1));

img3=figure;
plot(t(1:2001), v_lower(1:2001), 'b', 'LineWidth', 2);
```

```matlab
hold on;
plot(t(1:2001), v_higher(1:2001), 'r', 'LineWidth', 2);
hold on
plot(t(1:2001), v_lower_n(1:2001), 'g', 'LineWidth', 2);
hold on;
plot(t(1:2001), v_higher_n(1:2001), 'c', 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Membrane Potential (V)');
title('Membrane Potential with Different Applied Currents in presense of
noise');
legend('slightly lower than','slightly higher than','slightly lower than-with
noise','slightly higher than-with noise','Location','SouthEast');
grid on;
saveas(img3, sprintf('Tutorial_2_1_question_2a_1.png'));


n_frates = zeros(length(i_0),length(sigma));
k=1;
l=1;
for sigma_I=sigma

for I_0 = i_0

    [I_app_f,v_f]  = n_Iapp(I_0, t, leak_potential, C, R, dt, threshold,
reset_potential,sigma_I);

    spike_count = 0;

    for n = 1:length(v_f)
        if v_f(n) == reset_potential;
            spike_count = spike_count + 1;
        end
    end
    rate = spike_count/2;
    n_frates(k,l) = rate;
    k=k+1;
end
k=1;
l=l+1;
end

f_n=zeros(length(i_0),length(sigma));
j=1;
w=1;
for sigma_I=sigma
for I_0 = i_0
    if ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - reset_potential) > 0) & ...
        ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - threshold) > 0)

        firing_rate = 1 / (Tau * log((leak_potential + R * Iapp(I_0, t,
leak_potential, C, R, dt, threshold, reset_potential) - reset_potential) /
...
```

```
                       (leak_potential + R * Iapp(I_0, t, leak_potential, C,
R, dt, threshold, reset_potential) - threshold)));

    end

    f_n(j,w) = firing_rate;
    j = j + 1;
end
j=1;
w=w+1
end

img4=figure
scatter(i_0, frates, 'SizeData', 52);
hold on
scatter(i_0, f, 'SizeData', 52);
hold on
scatter(i_0, n_frates(:,1), 'Marker', '*', 'SizeData', 52);
hold on
scatter(i_0, f_n(:,1), 'Marker', '*', 'SizeData', 52);
hold on
scatter(i_0, n_frates(:,2), 'Marker', 's', 'SizeData', 52);
hold on
scatter(i_0, f_n(:,2), 'Marker', 's', 'SizeData',52);
xlabel('I_app');
ylabel('Firing rate(Hz)');
title('Firing rate based on injected current');
legend('1c-sigma=0','1d-sigma=0','1c-sigma=0.1','1d-sigma=0.1','1c-
sigma=0.2','1d-sigma=0.2','Location','SouthEast');
grid on;
saveas(img4, sprintf('Tutorial_2_1_question_2a.png'));
%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dt = 0.01e-3;     % Time step (s)
T = 2;      % Maximum time (s)

% Create time vector
t = 0:dt:T;

% I define the I_app and the loop after that in Iapp function
I_th = (threshold  - leak_potential)/R;
I_lower = I_th*0.99;
[I_app_lower,v_lower] = Iapp(I_lower,
t,leak_potential,C,R,dt,threshold,reset_potential);
I_higher = I_th*1.01;
[I_app_higher,v_higher] = Iapp(I_higher,
t,leak_potential,C,R,dt,threshold,reset_potential);

img5=figure;
plot(t(1:2001), v_lower(1:2001), 'b', 'LineWidth', 2);
hold on;
plot(t(1:2001), v_higher(1:2001), 'r', 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Membrane Potential (V)');
```

```
title('Membrane Potential with Different Applied Currents');
legend('slightly lower than','slightly higher than','Location','SouthEast');
grid on;
saveas(img5, sprintf('Tutorial_2_1_question_2c_1.png'));
%Lower Current: I_low doesn't produce spikes, the calculation and simulation
confirm the minimum current threshold for spiking.
%Higher Current:I_high produces spikes, it verifies that currents above the
threshold trigger spiking behavior.
Tau = C*R;
i_0 = [I_th*1.05 I_th*1.1 I_th*1.15 I_th*1.2 I_th*1.25 I_th*1.30 I_th*1.25
I_th*1.35 I_th*1.4 I_th*1.45];
frates = zeros(size(i_0));
k=1;
for I_0 = i_0

    [I_app_f,v_f]  = Iapp(I_0, t, leak_potential, C, R, dt, threshold,
reset_potential);

    spike_count = 0;

    for n = 1:length(v_f)
        if v_f(n) == reset_potential;
            spike_count = spike_count + 1;
        end
    end
    rate = spike_count/2;
    frates(k) = rate;
    k=k+1;
end
img1=figure
scatter(i_0, frates);
xlabel('I_app');
ylabel('Firing rate(Hz)');
title('Firing rate based on injected current 1c');
legend('1c','Location','SouthEast');
grid on;
saveas(img1, sprintf('Tutorial_2_1_question_1c.png'));

f=zeros(size(i_0));
j=1;
for I_0 = i_0
    if ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - reset_potential) > 0) & ...
       ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - threshold) > 0)

        firing_rate = 1 / (Tau * log((leak_potential + R * Iapp(I_0, t,
leak_potential, C, R, dt, threshold, reset_potential) - reset_potential) /
...
                       (leak_potential + R * Iapp(I_0, t, leak_potential, C,
R, dt, threshold, reset_potential) - threshold)));

    end

    f(j) = firing_rate;
```

```
    j = j + 1;
end

img6=figure
scatter(i_0, frates);
hold on
scatter(i_0, f);
xlabel('I_app');
ylabel('Firing rate(Hz)');
title('Firing rate based on injected current');
legend('1c','1d','Location','SouthEast');
grid on;
saveas(img6, sprintf('Tutorial_2_1_question_2c_2.png'));
sigma=[0.1,0.2]

I_lower = I_th*0.99;
[I_app_lower_n,v_lower_n] = n_Iapp(I_lower,
t,leak_potential,C,R,dt,threshold,reset_potential,sigma(1));
I_higher = I_th*1.01;
[I_app_higher_n,v_higher_n] = n_Iapp(I_higher,
t,leak_potential,C,R,dt,threshold,reset_potential,sigma(1));

img7=figure;
plot(t(1:2001), v_lower(1:2001), 'b', 'LineWidth', 2);
hold on;
plot(t(1:2001), v_higher(1:2001), 'r', 'LineWidth', 2);
hold on
plot(t(1:2001), v_lower_n(1:2001), 'g', 'LineWidth', 2);
hold on;
plot(t(1:2001), v_higher_n(1:2001), 'c', 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Membrane Potential (V)');
title('Membrane Potential with Different Applied Currents in presense of
noise');
legend('slightly lower than','slightly higher than','slightly lower than-with
noise','slightly higher than-with noise','Location','SouthEast');
grid on;
saveas(img7, sprintf('Tutorial_2_1_question_2c_3.png'));


n_frates = zeros(length(i_0),length(sigma));
k=1;
l=1;
for sigma_I=sigma

for I_0 = i_0

    [I_app_f,v_f]  = n_Iapp(I_0, t, leak_potential, C, R, dt, threshold,
reset_potential,sigma_I);

    spike_count = 0;

    for n = 1:length(v_f)
        if v_f(n) == reset_potential;
            spike_count = spike_count + 1;
```

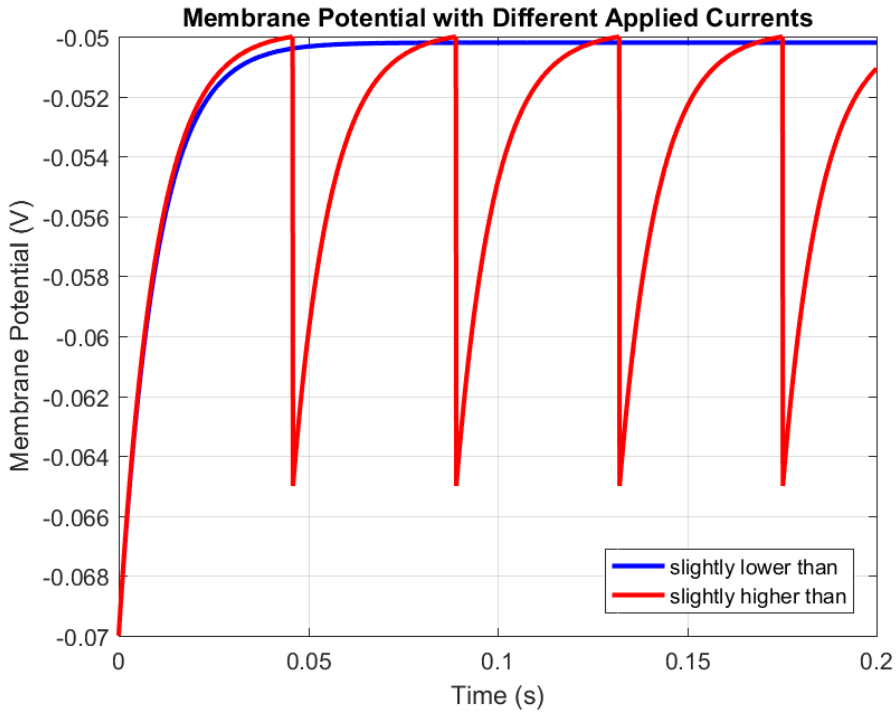
```
        end
    end
    rate = spike_count/2;
    n_frates(k,l) = rate;
    k=k+1;
end
k=1;
l=l+1;
end

f_n=zeros(length(i_0),length(sigma));
j=1;
w=1;
for sigma_I=sigma
for I_0 = i_0
    if ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - reset_potential) > 0) & ...
       ((leak_potential + R * Iapp(I_0, t, leak_potential, C, R, dt,
threshold, reset_potential) - threshold) > 0)

        firing_rate = 1 / (Tau * log((leak_potential + R * Iapp(I_0, t,
leak_potential, C, R, dt, threshold, reset_potential) - reset_potential) /
...
                       (leak_potential + R * Iapp(I_0, t, leak_potential, C,
R, dt, threshold, reset_potential) - threshold)));

    end

    f_n(j,w) = firing_rate;
    j = j + 1;
end
j=1;
w=w+1
end

img8=figure
scatter(i_0, frates, 'SizeData', 52);
hold on
scatter(i_0, f, 'SizeData', 52);
hold on
scatter(i_0, n_frates(:,1), 'Marker', '*', 'SizeData', 52);
hold on
scatter(i_0, f_n(:,1), 'Marker', '*', 'SizeData', 52);
hold on
scatter(i_0, n_frates(:,2), 'Marker', 's', 'SizeData', 52);
hold on
scatter(i_0, f_n(:,2), 'Marker', 's', 'SizeData',52);
xlabel('I_app');
ylabel('Firing rate(Hz)');
title('Firing rate based on injected current');
legend('1c-sigma=0','1d-sigma=0','1c-sigma=0.1','1d-sigma=0.1','1c-
sigma=0.2','1d-sigma=0.2','Location','SouthEast');
grid on;
saveas(img8, sprintf('Tutorial_2_1_question_2c_4.png'));
```

Functions I used in the code:

```
function [I_app,v] = Iapp(I_0,
t,leak_potential,C,R,dt,threshold,reset_potential)
    v =zeros(size(t));
    v(1) = leak_potential;
    I_app = I_0*ones(size(t));
    for i = 2:length(t)
        % Update membrane potential with Forward Euler
        v(i) = v(i-1) + dt * ((leak_potential-v(i-1)) / R + I_app(i)) / C;

        % Check for and reset spike
        if v(i) >= threshold
            v(i) = reset_potential;
        end
    end
end


function [I_app_n,v_n] = n_Iapp(I_0,
t,leak_potential,C,R,dt,threshold,reset_potential,sigma_I)
    v_n =zeros(size(t));
    v_n(1) = leak_potential;
    I_app_n = I_0*ones(size(t));
    noise_vec = randn(size(t))*sigma_I*sqrt(dt);
    for i = 2:length(t)
        % Update membrane potential with Forward Euler
        v_n(i) = (v_n(i-1) + dt * ((leak_potential-v_n(i-1)) / R +
I_app_n(i)) / C)+noise_vec(i);

        % Check for and reset spike
        if v_n(i) >= threshold
            v_n(i) = reset_potential;
        end
    end
end
```

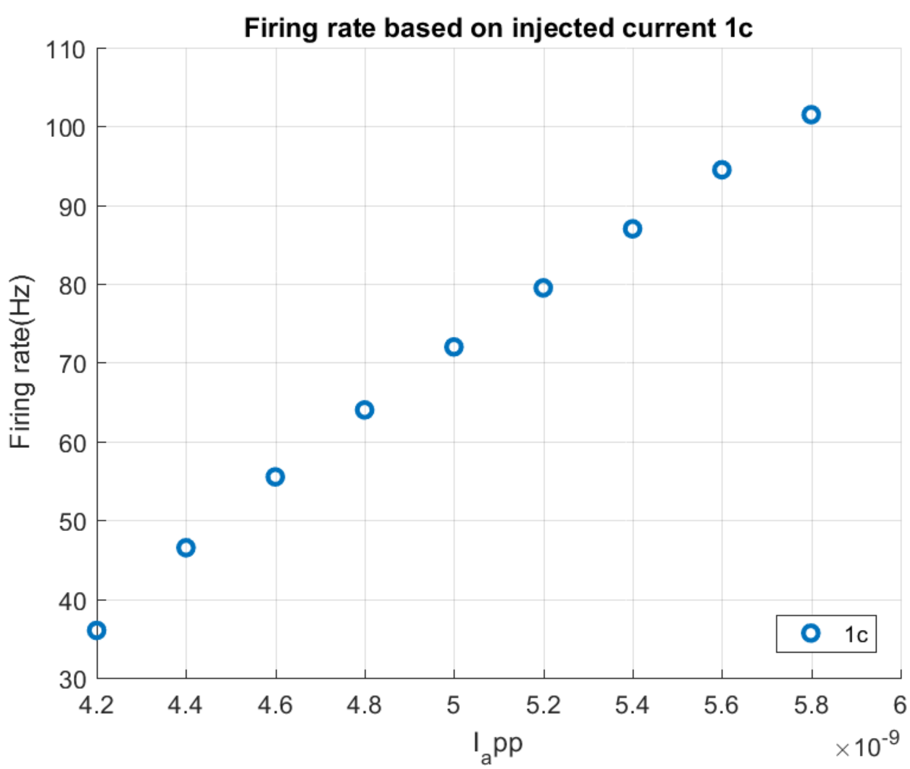
The simulation graphs:

1.

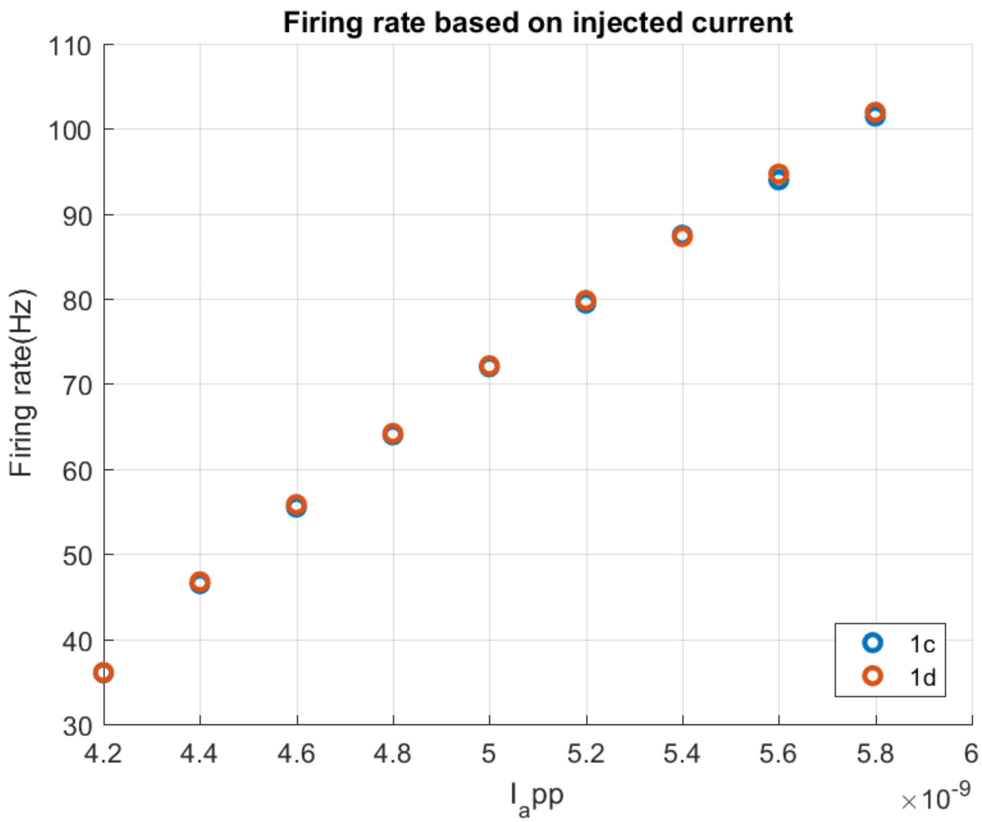


Lower Current: I_low doesn't produce spikes, the calculation and simulation confirm the minimum current threshold for spiking.
Higher Current:I_high produces spikes, it verifies that currents above the threshold trigger spiking behavior.

1.C.

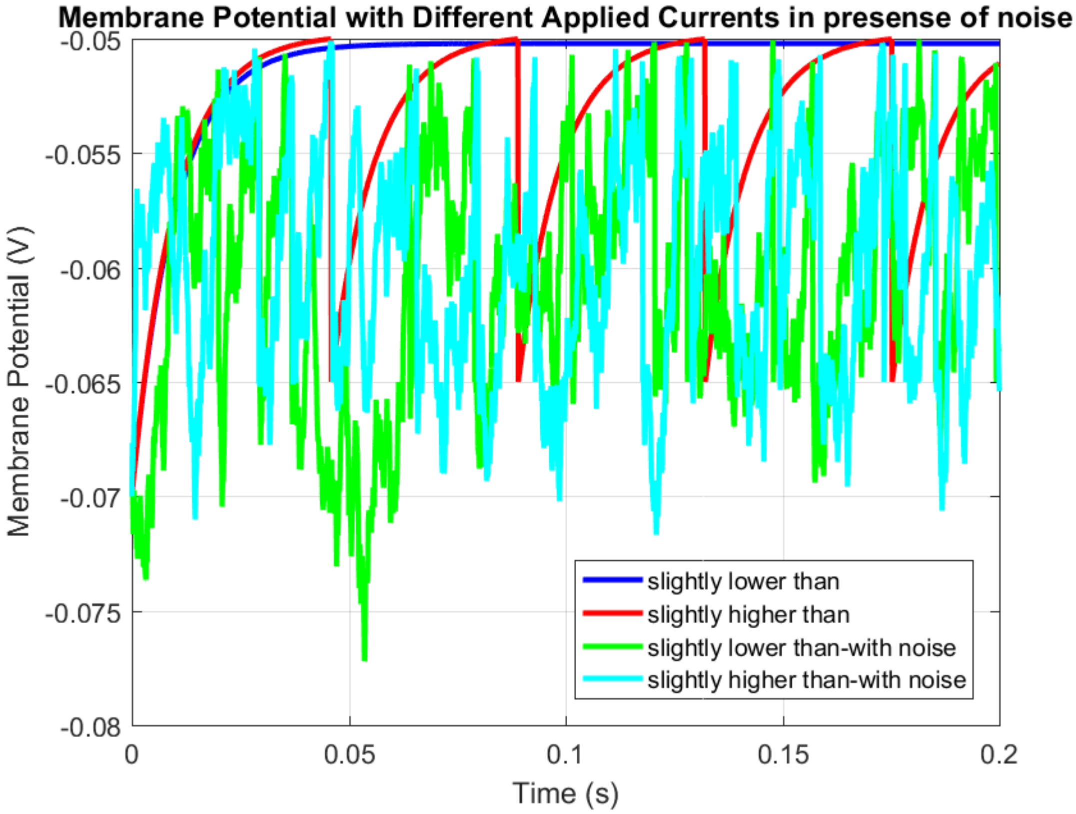


1.D.

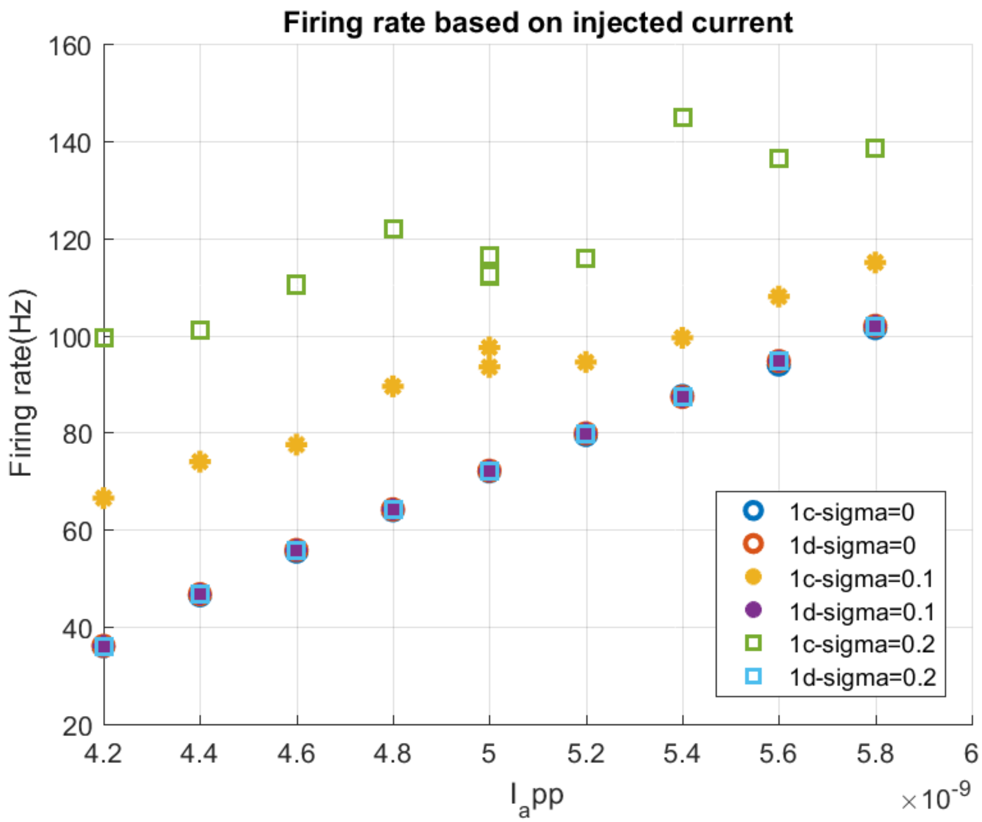


I can observe the different firing rates between the theoretical and estimated f_I curve.

2.a: adding noise.

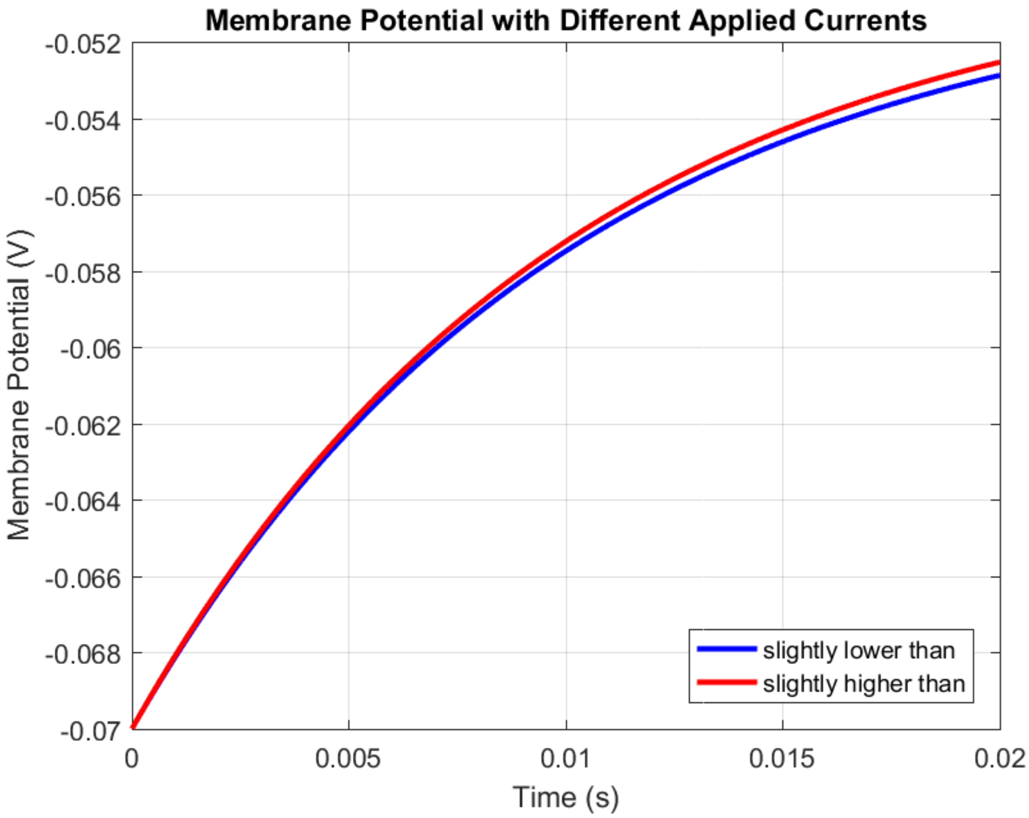


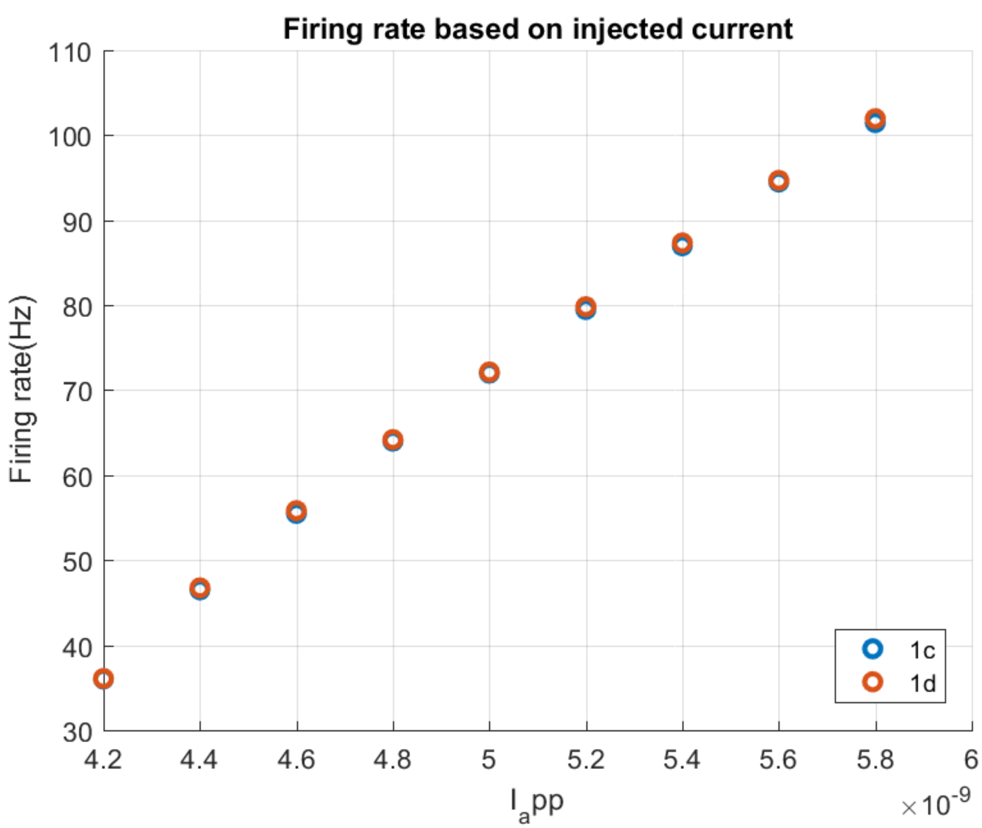
Adding noise with variance 0.1

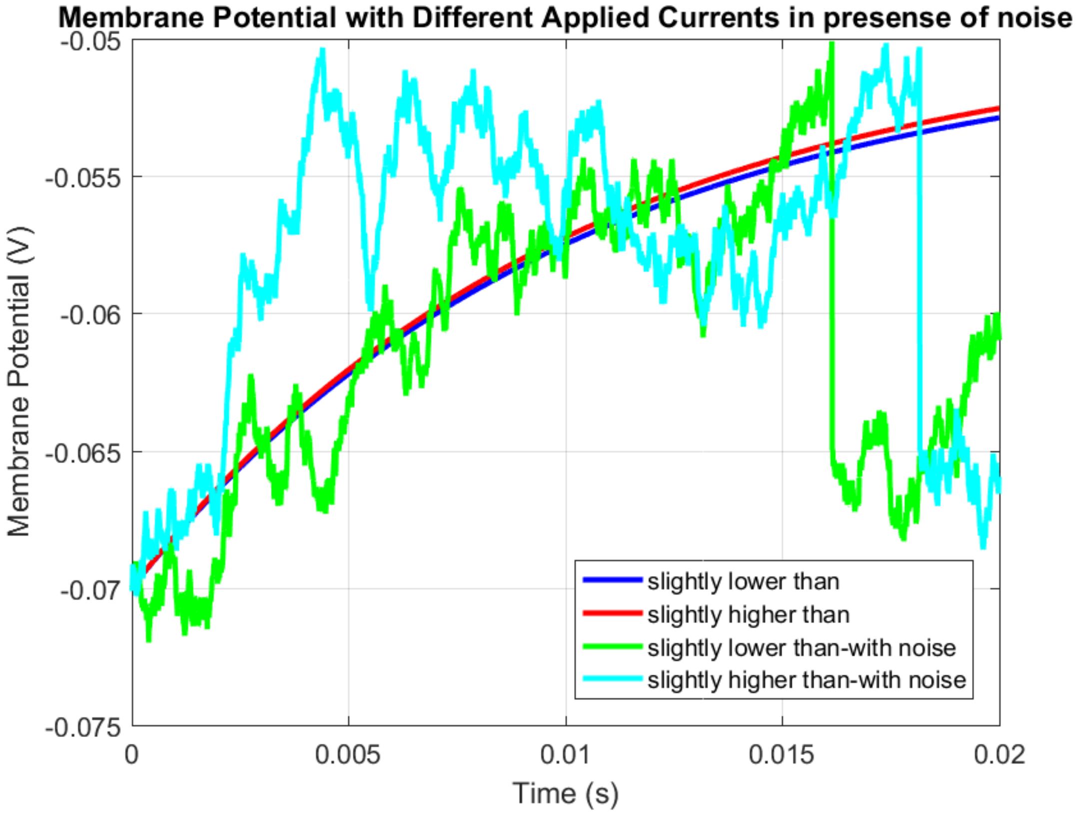


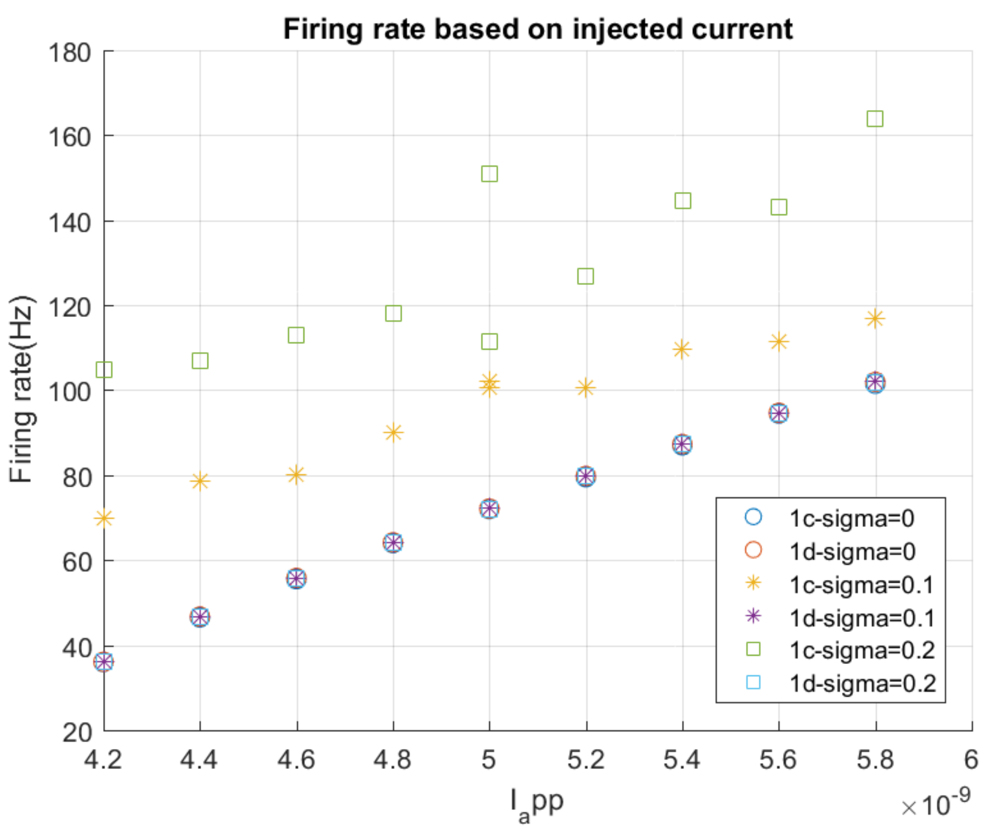
I can observe that the theoretical graph will be same in presence of noise but the estimation based on number of spikes will be change radically. also an increase in the parameter sigma, which is proportional to the standard deviation of the voltage noise, leads to an increase in the average firing rate when we use estimation based on number of spikes.

2c. changing dt to 0.01 ms:


Membrane Potential with Different Applied Currents
-0.052
-0.054
-0.056
-0.058
-0.06
-0.062
-0.064
-0.066
-0.068
-0.07
Membrane Potential (V)
0
0.005
0.01
0.015
0.02
Time (s)
slightly lower than
slightly higher than


Firing rate based on injected current
110
100
90
80
70
60
50
40
30
Firing rate(Hz)
4.2
4.4
4.6
4.8
5
5.2
5.4
5.6
5.8
6
$\times 10^{-9}$
$I_a$pp
1c
1d


Membrane Potential with Different Applied Currents in presense of noise
Membrane Potential (V)
Time (s)
slightly lower than
slightly higher than
slightly lower than-with noise
slightly higher than-with noise
-0.05
-0.055
-0.06
-0.065
-0.07
-0.075
0
0.005
0.01
0.015
0.02


Firing rate based on injected current
Firing rate(Hz)
I_app
×10^-9
1c-sigma=0
1d-sigma=0
1c-sigma=0.1
1d-sigma=0.1
1c-sigma=0.2
1d-sigma=0.2
180
160
140
120
100
80
60
40
20
4.2
4.4
4.6
4.8
5
5.2
5.4
5.6
5.8
6

As we can observe in graphs a finer timestep allows for a more accurate simulation of membrane potential changes