EEE 5283/EEL 4930: NEURAL SIGNALS, SYSTEMS, AND TECHNOLOGY

Spring 2019

Assignment 2
Due: Friday 02/01/2019 (11:59 PM)

Reading Assignment

- Under the 'Reading Material/week 3' folder, read the article 'How Good Are Neuron Models?' and 'Integrate and Fire Model of Spike Generation'

Problem 1

You are given the MATLAB code clamp.m that implements a voltage clamp experiment using the Hodgekin-Huxley Model

a) Modify the code to output ion current traces for voltage clamp values ranging between -50 mV and 75mV
   Take E_K= -89mV, E_{Na}= 55mV for a spherical cell with diameter 10µM
b) Compare the generated current traces to Figure 7-5 in Principles of Neuron Science Chapter 7. Are they similar? If yes, explain why. If no explain why not.
c) Assume that you were not provided E_K, E_{Na} or the membrane conductances. Describe a step by step approach to experimentally find them using a voltage clamp experiment and the current traces generated.

a)

```
%
% clamp.m
%
% Simulate a voltage clamp experiment
%
% usage: clamp(dt,Tfin)
%
% e.g. clamp(.01,15)
%
function clampI(dt,Tfin)
% Default parameters
if nargin ~= 2
dt = 0.1;
Tfin = 10;
end
EK = -89; % mV
GK = 36; % mS/(cm)^2
ENa = 55; % mV
GNa = 120; % mS/(cm)^2
% Cell Geometry
cell_d = 10e-6; %diameter in meters
cell_a = 4*pi*(cell_d/2)^2; %area in meters
vm = [-50 -30 -10 10 35 55 75];
for vc = vm,
j = 2;
t(1) = 0;
```

```
v(1) = 0;
n(1) = an(0)/(an(0)+bn(0)); % 0.3177;
m(1) = am(0)/(am(0)+bm(0)); % 0.0529;
h(1) = ah(0)/(ah(0)+bh(0)); % 0.5961;
gK(1) = GK*n(1)^4;
gNa(1) = GNa*m(1)^3*h(1);
iK(1) = gK(1) * (v(1) - EK)*cell_a;
iNa(1) = gNa(1) * (v(1) - ENa)*cell_a;
while j*dt < Tfin,
t(j) = j*dt;
v(j) = vc*(t(j)>2)*(t(j)<Tfin);
```

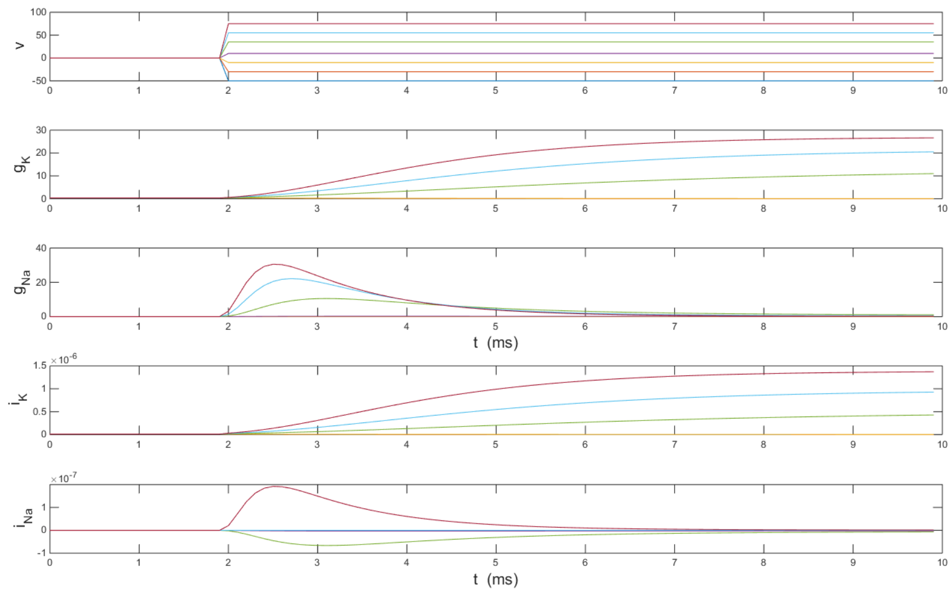


```
n(j) = ( n(j-1) + dt*an(v(j)) )/(1 + dt*(an(v(j))+bn(v(j))) );
m(j) = ( m(j-1) + dt*am(v(j)) )/(1 + dt*(am(v(j))+bm(v(j))) );
h(j) = ( h(j-1) + dt*ah(v(j)) )/(1 + dt*(ah(v(j))+bh(v(j))) );
gK(j) = GK*n(j)^4;
gNa(j) = GNa*m(j)^3*h(j);
iK(j) = gK(j) * (v(j) - EK)*cell_a;
iNa(j) = gNa(j) * (v(j) - ENa)*cell_a;
j = j + 1;
end
subplot(5,1,1); plot(t,v); hold on
subplot(5,1,2); plot(t,gK); hold on
subplot(5,1,3); plot(t,gNa); hold on
subplot(5,1,4); plot(t,iK); hold on
subplot(5,1,5); plot(t,iNa); hold on
%pause
end
subplot(5,1,1)
ylabel('v','fontsize',16)
hold off
subplot(5,1,2)
ylabel('g_K','fontsize',16)
hold off
subplot(5,1,3)
xlabel('t (ms)','fontsize',16)
ylabel('g_{Na}','fontsize',16)
hold off
subplot(5,1,4)
ylabel('i_K','fontsize',16)
hold off
subplot(5,1,5)
xlabel('t (ms)','fontsize',16)
ylabel('i_{Na}','fontsize',16)
hold off
% rate functions from page 519 of HH
% (with polarity switched to agree with modern usage)
function val = an(v)
val = .01*(10-v)./(exp(1-v/10)-1);
function val = bn(v)
val = .125*exp(-v/80);
function val = am(v)
val = .1*(25-v)./(exp(2.5-v/10)-1);
```



```
function val = bm(v)
val = 4*exp(-v/18);%4*exp(-v/18);
function val = ah(v)
```

```
val = 0.07*exp(-v/20);
function val = bh(v)
val = 1./(exp(3-v/10)+1);
```



b) Sample answers: Many solutions possible
Similarities:
Na has a fast response to depolarization
Na response to prolonged depolarization is to close
K has a slow response to depolarization
K response to prolonged depolarization is to remain open
Difference:
Magnitude of responses
c)
Lecture 4 slides for voltage clamp setup
Find Ek, ENa by finding the values of Vm for which Ik and INa reserve their polarities (Page 155 Ch7 in the textbook)
Calculate

$$g_k = \frac{I_k}{V_m - E_k}$$

$$g_{Na} = \frac{I_{Na}}{V_m - E_{Na}}$$

Problem 2

Properties of Axons

Dielectric constant κ = 7, electrical permittivity $\varepsilon_0 = 8.85 * 10^{-12}\ C^2/N*m^2$

a) Calculate the capacitance of an unmyelinated axon with length L= 1m, membrane thickness b= 10nm and radius a = 2.5 μm

Hint: $C = \frac{\kappa * \varepsilon_0 * S}{b}$ Where S is the surface area

The capacity of the axon is:

$$C = \frac{\kappa \cdot \epsilon_0 \cdot S}{b} = \frac{\kappa \cdot \epsilon_0}{b} \cdot (2\pi \cdot a \cdot L)$$

$$C = 9.7 \cdot 10^{-8} \frac{C^2}{N \cdot m} \approx 0.1\ \mu F$$

b) Calculate the membrane resistance and time constant given membrane electrical resistivity $R_m = 1.6\ 10^3\ \Omega cm^2$

$$r_m = \frac{R_m}{2\pi * a} = \frac{1.6 * 10^3\ [\Omega \text{cm}^2]}{2\pi * 2.5 * 10^{-4} [cm]} = 10^6\ \Omega \text{cm}$$

$$\tau_m = C_m R_m = 0.1\ \mu F * 10^6 \Omega \text{cm} = 10^{-7} F * 10^4 \Omega \text{m} = 1.0 * 10^{-3} s = 1\ ms$$

c) Calculate the space constant and signal speed given an axoplasm resistivity of $R_i = 50\ \Omega cm$

Hint: $v = \frac{\lambda}{\tau}$

$$r_i = \frac{R_i}{\pi * a^2} = \frac{50[\Omega \text{cm}]}{\pi * (2.5 * 10^{-4} [\text{cm}])^2} = 2.55 * 10^8 \frac{\Omega}{cm} = 2.55 * 10^{10} \frac{\Omega}{m}$$

$$\lambda = \sqrt{\frac{r_m}{r_i}} = \sqrt{\frac{10^6 [\Omega \text{cm}]}{2.55 * 10^8 \left[\frac{\Omega}{cm}\right]}} = \sqrt{\frac{1 cm^2}{255}} \cong \frac{1}{16} cm = \frac{1}{1600} m$$

κ != K. K was not given in the problem so velocity could not be determined. No points were taken off for this.

d) Calculate how long it takes a signal to travel the length of the axon. Does the signal velocity seem fast or slow?

$$\tau_{resp} = \frac{L}{K} = \frac{1m}{K\ m/s} = \frac{1}{K} s$$

Slow! Imagine if the simplest possible action or thought took 1.375 seconds to execute. (Did not deduct points for incorrect answer to fast/slow)

e) In myelinated axon fibers there exists a fixed relation between the radius of the axon and thickness of the myelin layer which simplifies the space constant equation to $\lambda = \sqrt{0.2 * a \quad * \frac{\text{Rm}}{\text{Ri}}}$

Compare the signal speed in a myelinated and unmyelinated axon of similar thickness. What is the radius of a myelinated axon with the same space constant as part c?

$\lambda = \sqrt{0.2 * a \quad * \frac{\mathrm{Rm}}{\mathrm{Ri}}}$ should read $\lambda = \sqrt{0.2 * \frac{1}{b} * a^2 * \frac{\mathrm{Rm}}{\mathrm{Ri}}}$

The numerical answer is unaffected, but units may be off.

$$\lambda = \sqrt{0.2 * a \quad * \frac{R_m}{R_i}} = \sqrt{0.2 * a \quad * \frac{1.6 * 10^3[\Omega \mathrm{cm}^2]}{50[\Omega \mathrm{cm}]}}$$

$$\lambda = \sqrt{16 * 10^6}\mathrm{cm}$$

$$\lambda = 4000 \text{ cm}$$

$$a = \frac{\frac{1}{16}}{4000} cm = 1.56 * 10^{-5} cm = 1.56 * 10^{-7} m = 156\ nm$$

$$a_{myelinated} \approx \frac{1}{16} a_{unmyelinated}$$

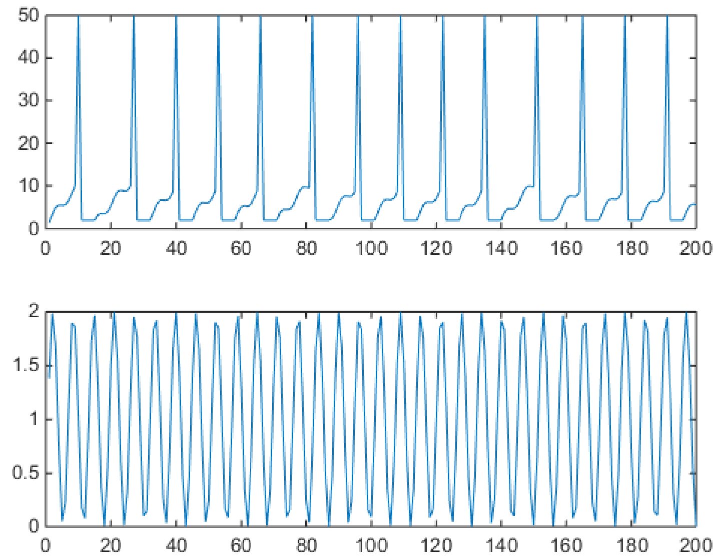
Problem 3

You are given MATLAB code intfire.m that implements a generic Integrate and Fire Model (IFM).

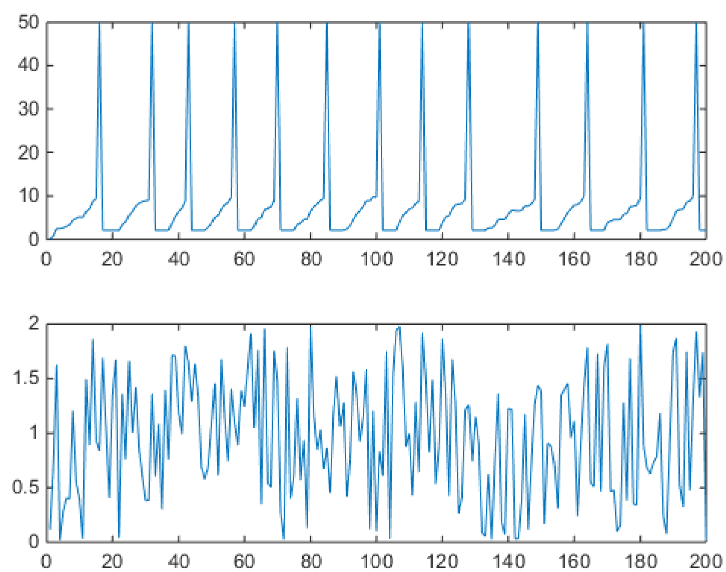
a) Change the constant current input to an offset sinusoid with the same average amplitude as the constant current input. Describe the model behavior and plot the results. Repeat the same experiment with a random amplitude input.

Sample answer:

```
I_p = sin(pi/8*1:tstop) + 1;
```




```
I_p = 2*rand(tstop,1)
```



b) Modify this model to implement equation 1 and use the following model parameters E_L =-70 mV, R_m =10 MΩ, and τ_m =10 ms. Initially set V =E_L.

Equation 1:

$$\tau_m \frac{dV}{dt} = E_L - V + R_M I_e$$

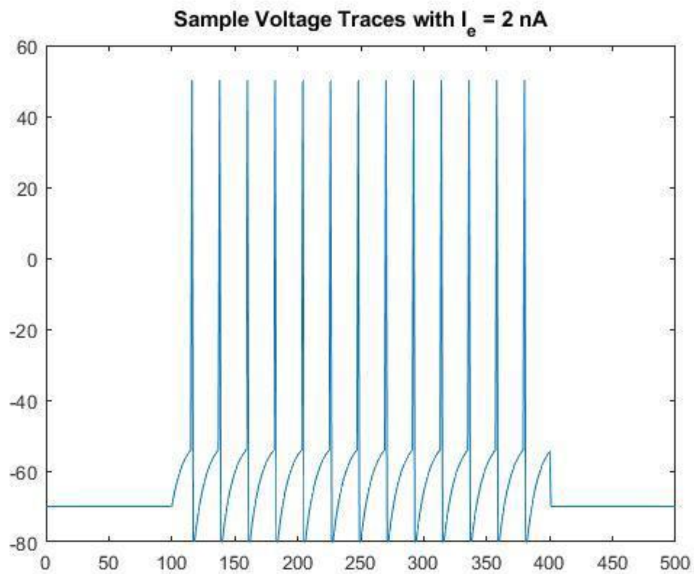
c) When the membrane potential reaches V_{th} = -54 mV, make the neuron fire a spike and reset the potential to V_{reset} = -80 mV. Show sample voltage traces (with spikes) for a 300-ms-long current pulse (choose a reasonable current I_e) centered in a 500-ms-long simulation.

d) Determine the firing rate of the model for various magnitudes of constant I_e and compare the results with equation 2.

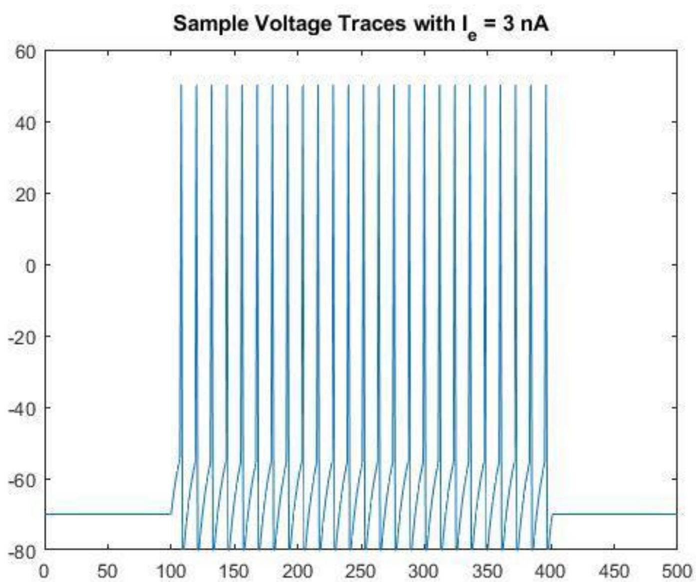
Equation 2

$$r_{\text{isi}} = \frac{1}{t_{\text{isi}}} = \left[\tau_m \ln\left(\frac{R_m I_e + E_L - V_{\text{reset}}}{R_m I_e + E_L - V_{\text{th}}}\right)\right]^{-1}$$

```
 % Basic integrate-and-fire neuron
clear,clc
% Iput: input current
I = 3; % nA
% Model parameters: capacitance and membrane resistance
% C = 1; % nF
R = 10; % M ohms
E = -70; % mV
tau = 10; % mS
% I & F implementation tau * dV/dt = E-V + R*I
% Using h = 1 ms step size, Euler method
V = E;
tstop = 300;
abs_ref = 2; % absolute refractory period
ref = 0; % absolute refractory period counter
V_trace = []; % voltage trace for plotting
V_th = -54; % spike threshold
V_reset = -80; %mv
count = 0; %number of spikes fired
for t = 1:tstop
if ~ref % If end of refractory period reached, % model voltage with I &
F equation.
V = V + ((E-V) + (R*I))/tau;
else % Else, in refractory period so hold voltage at reset level.
ref = ref - 1;
V = V_reset; % reset voltage
end
if (V > V_th) % If threshold voltage surpassed, fire an action
potential.
V = 50; % emit spike
ref = abs_ref; % set refractory counter
count = count + 1;
end
V_trace = [V_trace V];
end
V_trace = [E*ones(1,100),V_trace,E*ones(1,100)];
plot(V_trace)
title('Sample Voltage Traces with I_e = 3 nA')
r_isi = (tau*log((R*I+E-V_reset)/(R*I+E-V_th)))^-1*1000;
```

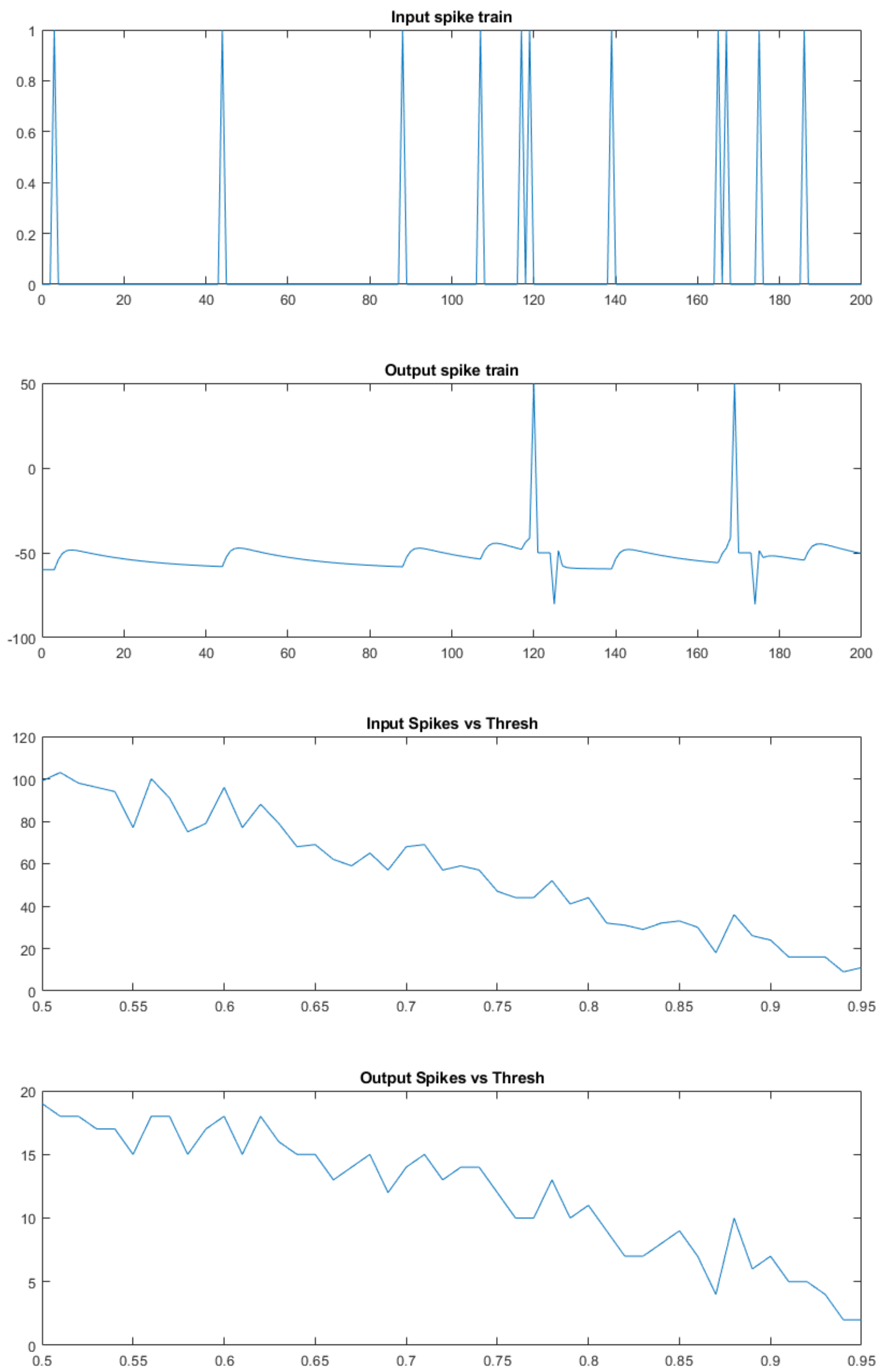


Sample Voltage Traces with I_e = 2 nA
60
40
20
0
-20
-40
-60
-80
0
50
100
150
200
250
300
350
400
450
500



Sample Voltage Traces with I_e = 3 nA
60
40
20
0
-20
-40
-60
-80
0
50
100
150
200
250
300
350
400
450
500

4.



b) Linear
c) Logarithmic (Exponential)

d) There were a few ways to change the conductance (g_ad, G_inc or alpha_func) No points taken off because changing the line "g_ad = 0" from the baseline code during a refractory period was not immediately obvious.

Sample solution changing G_inc:

```
% Fire a neuron via alpha function synapse and random input spike train

clear
rand('state',0)
% I & F implementation dV/dt = - V/RC + I/C
h = 1; % step size, Euler method, = dt ms
t_max= 200; % ms, simulation time period
tstop = t_max/h; % number of time steps
ref = 0; % refractory period counter

% Generate random input spikes
% Note: This is not entirely realistic - no refractory period
% Also: if you change step size h, input spike train changes too...
thr = [0.5:0.01:.95]; % threshold for random spikes
for i = 1:length(thr)
    spike_train = rand(tstop,1);
    spike_train(find(spike_train > thr(i))) =
ones(size((find(spike_train > thr(i)))));
    spike_train(find(spike_train < thr(i))) =
zeros(size((find(spike_train < thr(i)))));

    % alpha func synaptic conductance
    t_a = 100; % Max duration of syn conductance
    t_peak = 1; % ms
    g_peak = 0.05; % nS (peak synaptic conductance)
    const = g_peak/(t_peak*exp(-1));
    t_vec = 0:h:t_a;
    alpha_func = const*t_vec.*(exp(-t_vec/t_peak));
    clf
    plot(t_vec(1:80),alpha_func(1:80))
    xlabel('t (in ms)')
    title('Alpha Function (Synaptic Conductance for Spike at t=0)')

    % capacitance and leak resistance
    C = 0.5 % nF
    R = 40 % M ohms

    % Part II
    % conductance and associated parameters to simulate spike rate
adaptation

    G_inc = 0:0.25:1.5;
    for j = 1:length(G_inc)
        g_ad = 0;
        tau_ad = 2;

        % Initialize basic parameters
        E_leak = -60; % mV, equilibrium potential
        E_syn = 0; % Excitatory synapse (why is this excitatory?)
```

```
        g_syn = 0; % Current syn conductance
        V_th = -40; % spike threshold mV
        V_spike = 50; % spike value mV
        ref_max = 4/h; % Starting value of ref period counter
        t_list = [];
        V = E_leak;
        V_trace = [V];
        t_trace = [0];

        clf
        subplot(4,1,1)
        plot(0:h:t_max,[0; spike_train])
        title('Input spike train')

        for t = 1:tstop

            % Compute input
            if (spike_train(t) > 0) % check for input spike
                t_list = [t_list; 1];
            end
            % Calculate synaptic current due to current and past input
spikes
            g_syn = sum(alpha_func(t_list));
            I_syn = g_syn*(E_syn - V);

            % Update spike times
            if t_list
                t_list = t_list + 1;
                if (t_list(1) == t_a)  % Reached max duration of syn
conductance
                    t_list = t_list(2:max(size(t_list)));
                end
            end

            % Compute membrane voltage
            % Euler method: V(t+h) = V(t) + h*dV/dt
            if ~ref
                V = V + h*(- ((V-E_leak)*(1+R*g_ad)/(R*C)) +
(I_syn/C));
                g_ad = g_ad + h*(- g_ad/tau_ad); % spike rate
adaptation
            else
                ref = ref - 1;
                V = V_th-10; % reset voltage after spike
                %g_ad = 0;
            end

            % Generate spike
            if ((V > V_th) & ~ref)
                V = V_spike;
                ref = ref_max;
                g_ad = g_ad + G_inc(j);
            end

            V_trace = [V_trace V];
```

```
            t_trace = [t_trace t*h];
        end

        spike_count_in(i,j) = sum(spike_train);
        spike_count_out(i,j) = sum(V_trace==V_spike);
    end
end
subplot(4,1,2)
plot(t_trace,V_trace)
title('Output spike train')
subplot(4,1,3)
plot(thr,spike_count_in(:,5));
title('Input Spikes vs Thresh')
subplot(4,1,4)
plot(thr,spike_count_out(:,5));
title('Output Spikes vs Thresh')

drawnow

figure
hold on
for j = 1:length(G_inc)
    scatter(spike_count_in(:,j),spike_count_out(:,j))
end
legend(num2str(G_inc'))
title('Input vs Output Spikes @ Conductance')
hold off
```