Question 2

Question 3

Question 4

% From Question

Vrest = -70E-3; % Resting Voltage

R\_m = 10E6; % Membrane Resistance

tao = 10E-3; % Time Constant of membrane

Vm = -70E-3; % Membrane Voltage

Vth = -54E-3; % Threshold Voltage

Vreset = -80E-3; % Reset Voltage once spiked

time = 0.5; % Duration of simulation (s)

% Chosen values

Vspike = 0; % Spiking voltage

dt = 1E-3; % Time Step

I\_max = 2E-9; % Applied current

% Number of timesteps

iterations = time/dt;

% Create current vector

I = zeros(iterations, 1);

% Apply step function current for middle 300ms

I(0.2\* iterations + 1 : 0.8 \* iterations) = I\_max;

% iterate through phase timesteps

for n= 1:iterations

% Manual Spikes

if Vm >= Vth

% Add Spike to plot vector but reset to Vreset underneath

V(n, 1) = Vspike;

Vm = Vreset;

% Use the membrane equation

else

% Update Vm from membrane equation, add to plot vector

dv = (1/tao) \* ((-1\* Vm) + Vrest + R\_m\*I(n,1)) \* dt;

Vm = Vm + dv; % Euler increment Vm

V(n, 1) = Vm; % add to plot vector

end

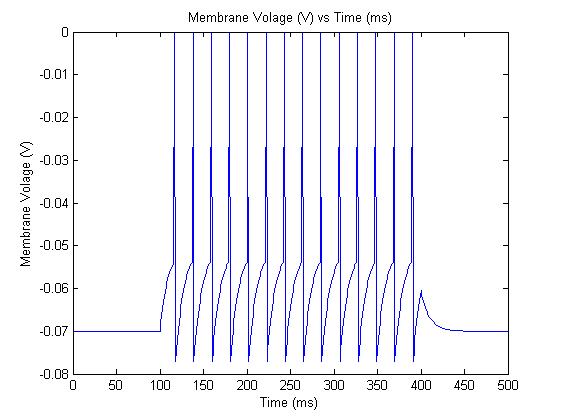
end

plot(V);

xlabel('Time (ms)')

ylabel('Membrane Volage (V)')

title('Membrane Volage (V) vs Time (ms)')



Question 5

% 0 to 10 ns in 100 increments

Iinj = 0:(10E-9/100):10E-9;

% Apply the simulated and expected spike rates for all currents

spike\_rate\_actual = arrayfun(@ass\_1\_q5\_actual, Iinj);

spike\_rate\_expected = arrayfun(@ass\_1\_q5\_expect, Iinj);

hold on

plot(Iinj, spike\_rate\_actual);

plot(Iinj, spike\_rate\_expected);

xlabel('Injection current, (A)')

ylabel('firing rate (/s)')

title('Firing Rate (/s) vs Injection current (A)')

hold off

with functions defined:

function firing\_rate = ass\_1\_q5\_actual( I\_inj)

% From Question

Vrest = -70E-3; % Resting Voltage

R\_m = 10E6; % Membrane Resistance

tao = 10E-3; % Time Constant of membrane

Vm = -70E-3; % Membrane Voltage

Vth = -54E-3; % Threshold Voltage

Vreset = -80E-3; % Reset Voltage once spiked

time = 0.5; % Duration of simulation (s)

% Chosen values

Vspike = 0; % Spiking voltage

dt = 1E-3; % Time Step

spikes = 0; % Tally of spikes triggered in simulation

% Number of timesteps

iterations = time/dt;

% Create current vector

I = zeros(iterations, 1);

% Apply step function current for middle 300ms

I(0.2\* iterations + 1 : 0.8 \* iterations) = I\_inj;

% iterate through phase timesteps

for n= 1:iterations

% Manual Spikes

if Vm >= Vth

% Add Spike to plot vector but reset to Vreset underneath

V(n, 1) = Vspike;

Vm = Vreset;

% Update number of spikes that have occured

spikes = spikes + 1;

% Use the membrane equation

else

% Update Vm from membrane equation

dv = (1/tao) \* ((-1\* Vm) + Vrest + R\_m\*I(n,1)) \* dt;

Vm = Vm + dv; % Euler increment Vm

V(n, 1) = Vm; % add to plot vector

end

end

% firing rate is spikes/time where time was only 60% of full simulation

firing\_rate = spikes / (0.6 \* time);

end

and

function firing\_rate = ass\_1\_q5\_expect( Iinj )

% From Question

Vrest = -70E-3; % Resting Voltage

R\_m = 10E6; % Membrane Resistance

tao = 10E-3; % Time Constant of membrane

Vm = -70E-3; % Membrane Voltage

Vth = -54E-3; % Threshold Voltage

Vreset = -80E-3; % Reset Voltage once spiked

time = 0.5; % Duration of simulation (s)

% Chosen values

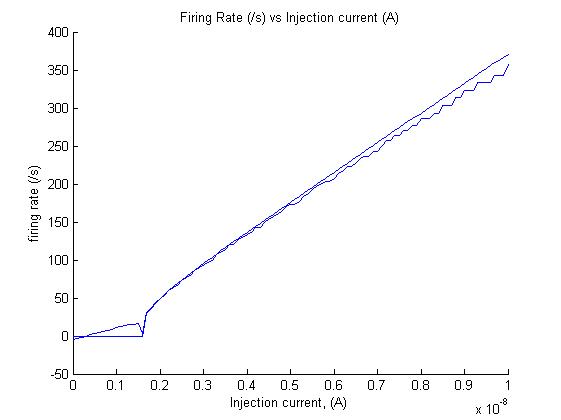
Vspike = 0; % Spiking voltage

dt = 1E-4; % Time Step

% Apply formula from assignment

firing\_rate = 1 /(tao \* log( (R\_m \* Iinj + Vrest - Vreset) / (R\_m \* Iinj + Vrest - Vth)) );

end



Question 6

% From Question

Vrest = -70E-3; % Resting Voltage

R\_m = 10E6; % Membrane Resistance

tao = 10E-3; % Time Constant of membrane

Vm = -70E-3; % Membrane Voltage

Vth = -54E-3; % Threshold Voltage

Vreset = -80E-3; % Reset Voltage once spiked

time = 0.5; % Duration of simulation (s)

% Chosen values

Vspike = 0; % Spiking voltage

dt = 1E-3; % Time Step

I\_max = 2E-9; % Applied current

% sra value from question

tao\_sra = 100E-3; % Time Constant for Spike rate Adaptation

dg\_sra = 0.3; % SRA increment value

V\_k = -70E-3; % Comparison voltage for SRA (equilibrium point)

r\_sra = 1; % Resistance value for SRA (considered in tao\_sra)

g\_sra\_init = 0; % initial value for g

g\_sra = g\_sra\_init; % Current g value in Euler iterations

iterations = time/dt;

% Create current vector

I = zeros(iterations, 1);

% Apply step function current for middle 300ms

I(0.2\* iterations + 1 : 0.8 \* iterations) = I\_max;

% iterate through phase timesteps

for n= 1:iterations

% Manual Spikes

if Vm >= Vth

% Set Spike in plot vector but reset voltage underneath

V(n, 1) = Vspike; % Add spike to plot vector

Vm = Vreset;

% increment g\_sra

g\_sra = g\_sra + dg\_sra;

% Use the membrane equation

else

% Update Vm via equation

dv = (1/tao) \* ((-1\* Vm) + Vrest + R\_m\*I(n,1) - g\_sra \* r\_sra \* (Vm - V\_k) ) \* dt ;

Vm = Vm + dv; % Euler Increment

% Spike rate adaptation variable updates

dg = (-1 \* g\_sra/tao\_sra) \* dt;

g\_sra = g\_sra + dg; % add sra decay to g\_sra

V(n, 1) = Vm; % Add voltage to plot vector

end

end

