close all

clear

clc

G\_Leak = 30E-9; % Leak conductance (S)

R\_m = 1/G\_Leak;% resistance (ohm)

G\_Na = 12E-6; %sodium conductance

G\_K = 3.6E-6; %potassium conductance

E\_Na = 45E-3;

E\_K = -82E-3;

leak\_potential = -60E-3;

C\_m = 100E-12;

tau\_m = R\_m\*C\_m;

dt = 0.02E-4;

tmax = 0.35;

time\_vector =0:dt:tmax;

v = zeros(size(time\_vector));

v(1) = leak\_potential;

%part a

applied\_current = zeros(size(time\_vector));

%initialize gating variables

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0;

h(1) = 0;

n(1) = 0;

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

f1=figure;

plot(time\_vector, v);

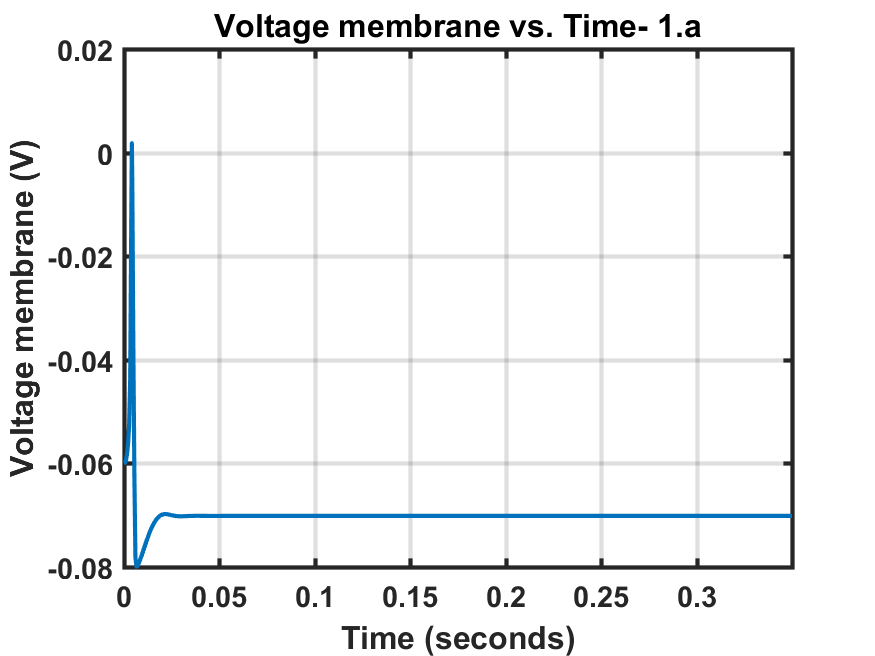
xlabel('Time (seconds)');

ylabel('Voltage membrane (V)');

title('Voltage membrane vs. Time- 1.a');

grid on

saveas(f1, sprintf('1\_a.png'));



%% part b

% Create applied current vector with step

v = zeros(size(time\_vector));

v(1) = leak\_potential;

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0;

h(1) = 0;

n(1) = 0;

applied\_current(time\_vector >= 100e-3 & time\_vector < 200e-3) = 0.22e-9;

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

% Plot the applied current

f2=figure;

subplot(2,1,1);

plot(time\_vector\*1e3, applied\_current\*1e9)

xlabel('Time (ms)')

ylabel('Applied Current (nA)')

title('Applied Current with Step - 1b')

grid on

subplot(2,1,2);

plot(time\_vector\*1e3, v\*1e3)

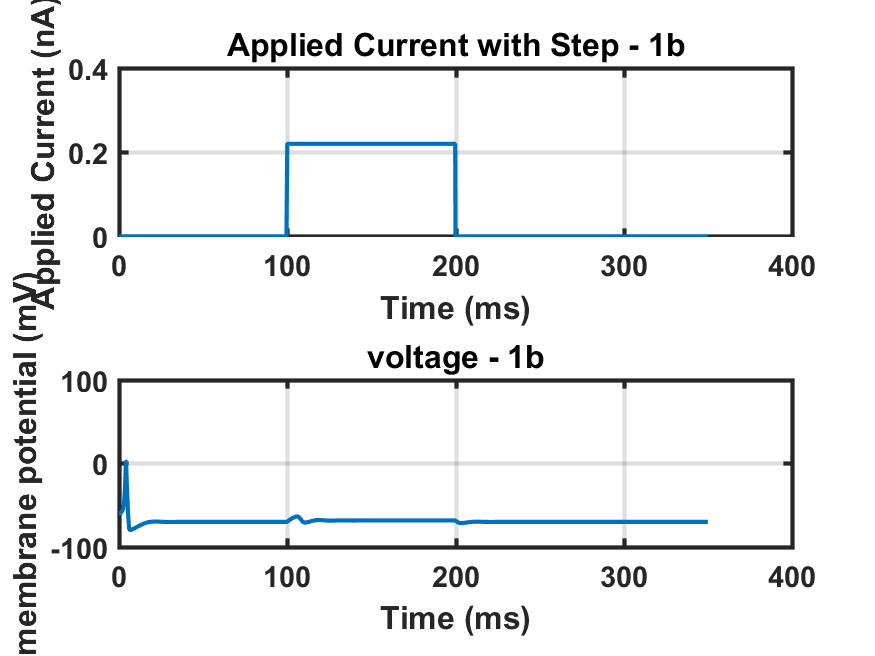
xlabel('Time (ms)')

ylabel('membrane potential (mV)')

title('voltage - 1b')

grid on

saveas(f2, sprintf('1\_b.png'));



With this input, the membrane potential undergoes subthreshold oscillations but doesn't reach the threshold for generating action potentials (spikes).

%%Part c

applied\_current = zeros(size(time\_vector));

duration=5e-3/dt;

delay=0;

for j=1:10

startpoint=((100e-3/dt)+delay+duration\*(j-1));

endpoint=(startpoint+duration);

applied\_current(startpoint: endpoint) = 0.22e-9;

delay=j\*5e-3/dt;

if startpoint>=200e-3/dt

applied\_current(startpoint:endpoint)=0;

end

end

v = zeros(size(time\_vector));

v(1) = leak\_potential;

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0;

h(1) = 0;

n(1) = 0;

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

f3=figure;

subplot(2,1,1);

plot(time\_vector\*1e3, applied\_current\*1e9)

xlabel('Time (ms)')

ylabel('Applied Current (nA)')

title('Applied Current with Step - 1c - 5ms delay')

grid on

subplot(2,1,2);

plot(time\_vector\*1e3, v\*1e3)

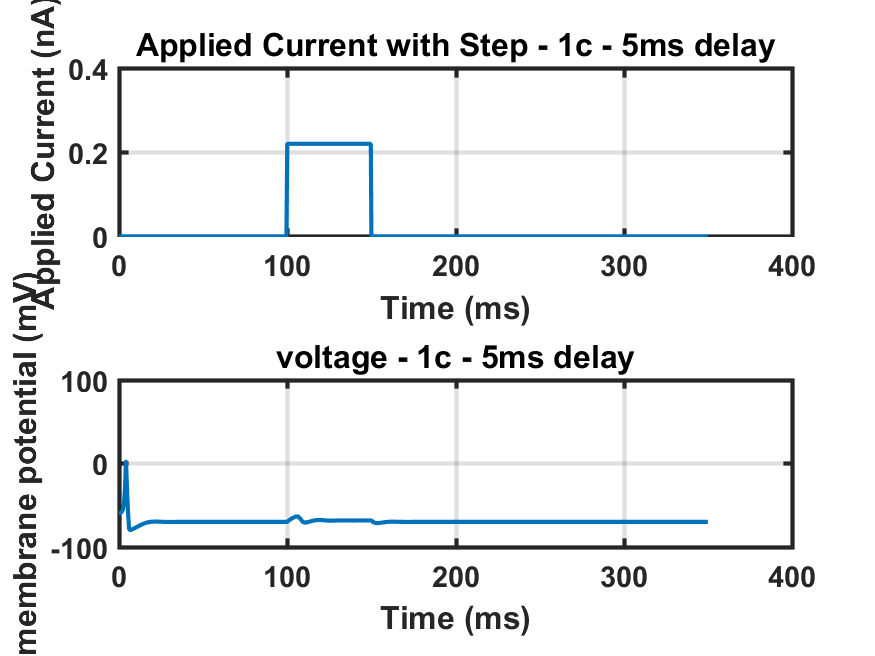
xlabel('Time (ms)')

ylabel('membrane potential (mV)')

title('voltage - 1c - 5ms delay')

grid on

saveas(f3, sprintf('1\_c-5ms delay.png'));



Here bistability occurs because we have changed the input but still getting same answer (two different inputs give same response)

applied\_current = zeros(size(time\_vector));

duration=5e-3/dt;

delay=0;

for j=1:10

startpoint=((100e-3/dt)+delay+duration\*(j-1));

endpoint=(startpoint+duration);

applied\_current(startpoint: endpoint) = 0.22e-9;

delay=j\*10e-3/dt;

if startpoint>=200e-3/dt

applied\_current(startpoint:endpoint)=0;

end

end

v = zeros(size(time\_vector));

v(1) = leak\_potential;

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0;

h(1) = 0;

n(1) = 0;

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

f4=figure;

subplot(2,1,1);

plot(time\_vector\*1e3, applied\_current\*1e9)

xlabel('Time (ms)')

ylabel('Applied Current (nA)')

title('Applied Current with Step - 1c - 10ms delay')

grid on

subplot(2,1,2);

plot(time\_vector\*1e3, v\*1e3)

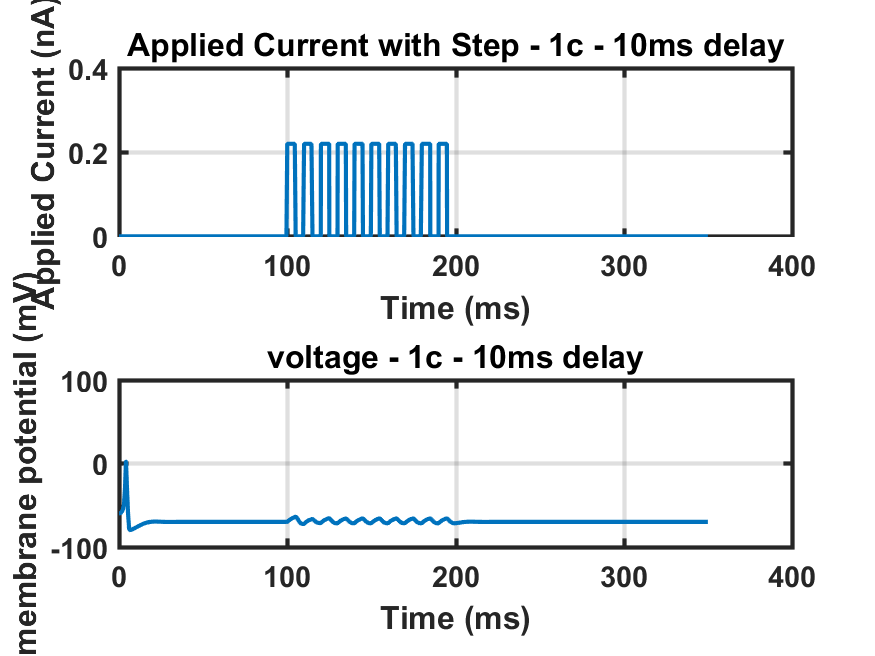
xlabel('Time (ms)')

ylabel('membrane potential (mV)')

title('voltage - 1c - 10ms delay')

grid on

saveas(f4, sprintf('1\_c-10ms delay.png'));



applied\_current = zeros(size(time\_vector));

duration=5e-3/dt;

delay=0;

for j=1:10

startpoint=((100e-3/dt)+delay+duration\*(j-1));

endpoint=(startpoint+duration);

applied\_current(startpoint: endpoint) = 0.22e-9;

delay=j\*15e-3/dt;

if startpoint>=200e-3/dt

applied\_current(startpoint:endpoint)=0;

end

end

v = zeros(size(time\_vector));

v(1) = leak\_potential;

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0;

h(1) = 0;

n(1) = 0;

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

f5=figure;

subplot(2,1,1);

plot(time\_vector\*1e3, applied\_current\*1e9)

xlabel('Time (ms)')

ylabel('Applied Current (nA)')

title('Applied Current with Step - 1c - 15ms delay')

grid on

subplot(2,1,2);

plot(time\_vector\*1e3, v\*1e3)

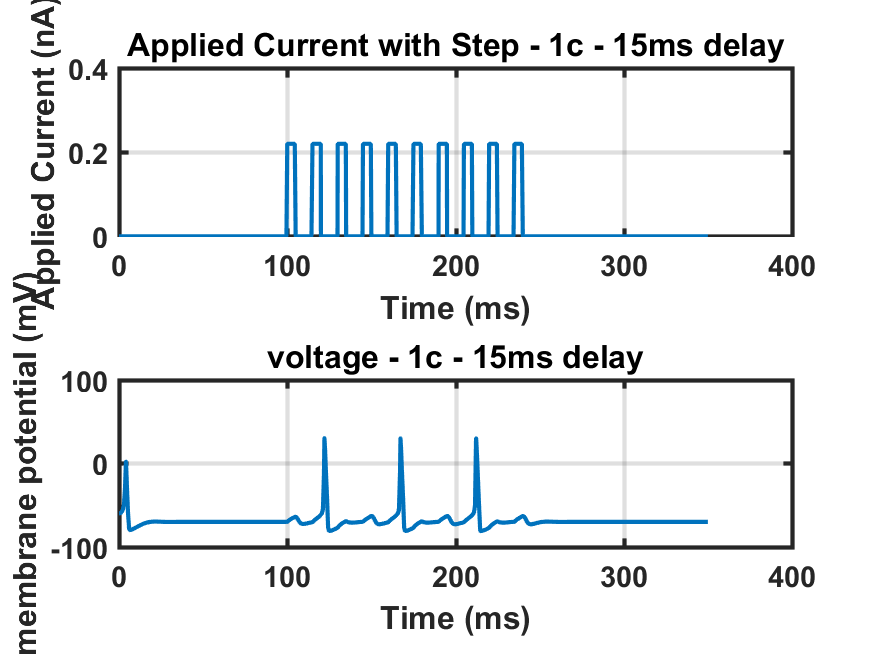
xlabel('Time (ms)')

ylabel('membrane potential (mV)')

title('voltage - 1c - 15ms delay')

grid on

saveas(f5, sprintf('1\_c-15ms delay.png'));



%Below a certain threshold: The delay might be too short for the sodium channels to recover from inactivation, preventing spike generation.

%Optimal range: There might be a "sweet spot" where the delay allows enough sodium channel recovery but keeps the membrane potential sufficiently depolarized for a strong response.

%Beyond a certain threshold: Very long delays might allow the membrane potential to return too close to the resting potential, making it less likely to reach the threshold even with high-amplitude, long-duration pulses

%Increased delay means less sodium channels are ready at the pulse onset, as they experience inactivation during the resting period. This initially reduces the inward sodium current, making it harder to reach the threshold.

%However, a longer delay also allows more time for the potassium channels to close, increasing the overall membrane resistance and promoting further depolarization.

%Depending on the interplay of these opposing effects, increasing the delay can sometimes lead to larger spikes due to greater membrane resistance if the sodium channels recover enough by the time the pulse arrives.

%Very long delays typically result in no spikes as the membrane potential returns closer to resting potential, requiring a stronger or longer pulse to overcome the leak current.

%Time-dependent inactivation: Sodium channels have an intrinsic inactivation mechanism. Once activated by a depolarization, they deactivate and enter an inactivated state over time. The recovery from this inactivation also takes time, following a specific rate constant.

%Leak current: During the resting period between pulses, the leak current counteracts depolarization and gradually pulls the membrane potential back towards its resting level.

%Interplay of recovery and leak: While a longer delay (like 15 ms) offers more time for recovery, it also allows the leak current more time to repolarize the membrane. This can counteract the sodium channel recovery, making it harder for the membrane potential to reach the threshold and trigger a spike.

%Why 10 ms might be an "optimal" delay in your case:

%With a 10 ms delay, the sodium channels might partially recover, but not completely. This can be enough to generate a significant inward current upon pulse arrival, pushing the membrane potential closer to the threshold.

%Additionally, the 10 ms delay might not give the leak current enough time to fully repolarize the membrane, leaving it still sufficiently depolarized for the recovered sodium channels to contribute a strong response.

%Why 15 ms might not trigger a spike:

%As the delay increases to 15 ms, the leak current has more time to act, potentially pulling the membrane potential further away from the threshold.

%Even though some sodium channels might recover from inactivation, the initial membrane potential could be too low to overcome the leak current and the remaining inactivated channels, preventing a strong enough depolarization to reach the threshold.

%% part D

applied\_current = 0.6e-9\*ones(size(time\_vector));

v = zeros(size(time\_vector));

v(1) = -0.065;

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0.05;

h(1) = 0.5;

n(1) = 0.35;

duration=5e-3/dt;

delay=0;

for j=1:10

startpoint=((100e-3/dt)+delay+duration\*(j-1));

endpoint=(startpoint+duration);

applied\_current(startpoint: endpoint) =0;

delay=j\*20e-3/dt;

end

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

f6=figure;

subplot(2,1,1);

plot(time\_vector\*1e3, applied\_current\*1e9)

xlabel('Time (ms)')

ylabel('Applied Current (nA)')

title('Applied Current with Step - 1d ')

grid on

subplot(2,1,2);

plot(time\_vector\*1e3, v\*1e3)

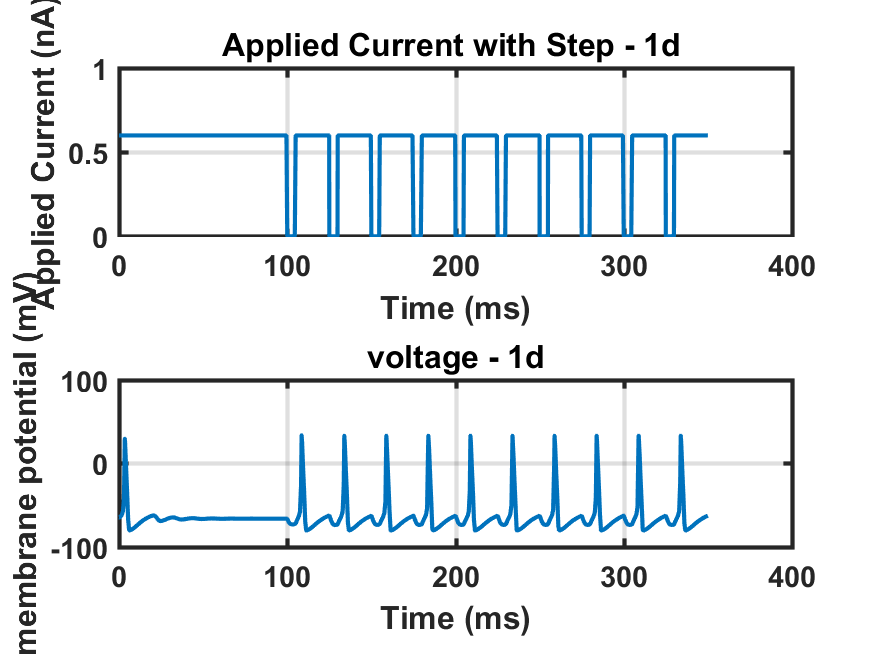
xlabel('Time (ms)')

ylabel('membrane potential (mV)')

title('voltage - 1d')

grid on

saveas(f6, sprintf('1\_d.png'));



%I see that

%% part e

applied\_current = 0.65e-9\*ones(size(time\_vector));

v = zeros(size(time\_vector));

v(1) = -0.065;

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0.05;

h(1) = 0.5;

n(1) = 0.35;

duration=5e-3/dt;

delay=0;

for j=1

startpoint=((100e-3/dt)+delay+duration\*(j-1));

endpoint=(startpoint+duration);

applied\_current(startpoint: endpoint) =1e-9;

delay=j\*20e-3/dt;

end

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

f7=figure;

subplot(2,1,1);

plot(time\_vector\*1e3, applied\_current\*1e9)

xlabel('Time (ms)')

ylabel('Applied Current (nA)')

title('Applied Current with Step - 1e ')

grid on

subplot(2,1,2);

plot(time\_vector\*1e3, v\*1e3)

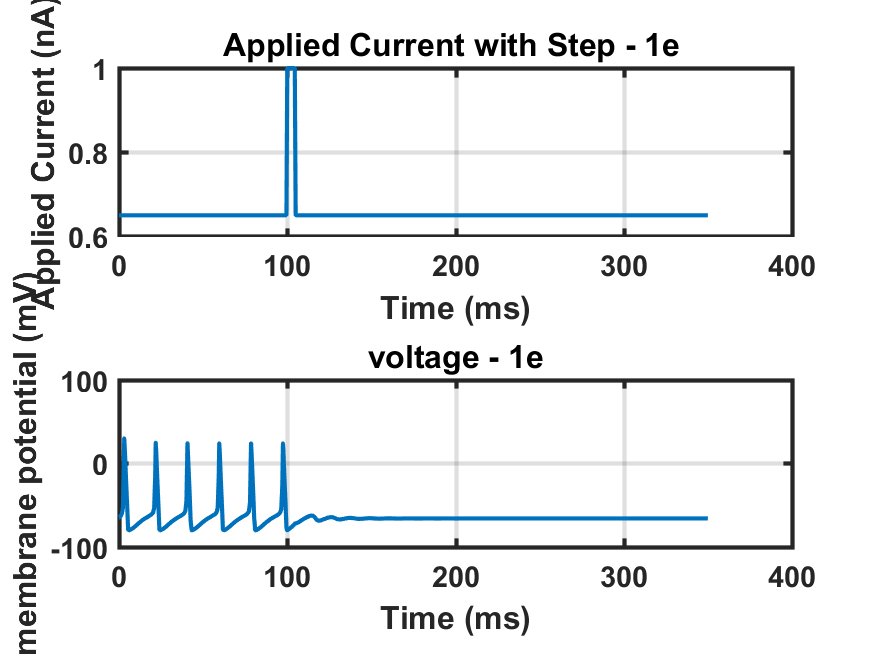
xlabel('Time (ms)')

ylabel('membrane potential (mV)')

title('voltage - 1e')

grid on

saveas(f7, sprintf('1\_e.png'));



Spiking resonance is ended just by applying the a spike at 100ms

%% part f

applied\_current = 0.7e-9\*ones(size(time\_vector));

v = zeros(size(time\_vector));

v(1) = -0.065;

m = zeros(size(time\_vector));

h = zeros(size(time\_vector));

n = zeros(size(time\_vector));

m(1) = 0;

h(1) =0;

n(1) = 0;

duration=5e-3/dt;

delay=0;

for j=1

startpoint=((100e-3/dt)+delay+duration\*(j-1));

endpoint=(startpoint+duration);

applied\_current(startpoint: endpoint) =1e-9;

delay=j\*20e-3/dt;

end

for j = 1:length(time\_vector)-1

v(j+1) = v(j) + dt\*( G\_Leak\*(leak\_potential-v(j)) + G\_Na\*(m(j)^3)\*h(j)\*(E\_Na-v(j))+G\_K\*(n(j)^4)\*(E\_K-v(j))+ applied\_current(j))/C\_m;

if ( v(j) == -0.045 ) % to avoid dividing zero by zero

alpha\_m = 1e3; % value calculated analytically

else

alpha\_m=(1e5\*(-v(j)-0.045))/(exp(100\*(-v(j)-0.045))-1);

end

beta\_m=4e3\*exp((-v(j)-0.070)/0.018);

alpha\_h=70\*exp(50\*(-v(j)-0.070));

beta\_h=1e3/(1+exp(100\*(-v(j)-0.040)));

if ( v(j) == -0.060) % to avoid dividing by zero

alpha\_n = 100; % value calculated analytically

else; % potassium activation rate

alpha\_n=(1e4\*(-v(j)-0.060))/(exp(100\*(-v(j)-0.060))-1);

end

beta\_n=125\*exp((-v(j)-0.070)/0.08);

m(j+1)=m(j)+dt\*(alpha\_m\*(1-m(j))-beta\_m\*m(j) );

h(j+1)=h(j)+dt\*(alpha\_h\*(1-h(j))-beta\_h\*h(j));

n(j+1)=n(j)+dt\*(alpha\_n\*(1-n(j))-beta\_n\*n(j));

end

f8=figure;

subplot(2,1,1);

plot(time\_vector\*1e3, applied\_current\*1e9)

xlabel('Time (ms)')

ylabel('Applied Current (nA)')

title('Applied Current with Step - 1f ')

grid on

subplot(2,1,2);

plot(time\_vector\*1e3, v\*1e3)

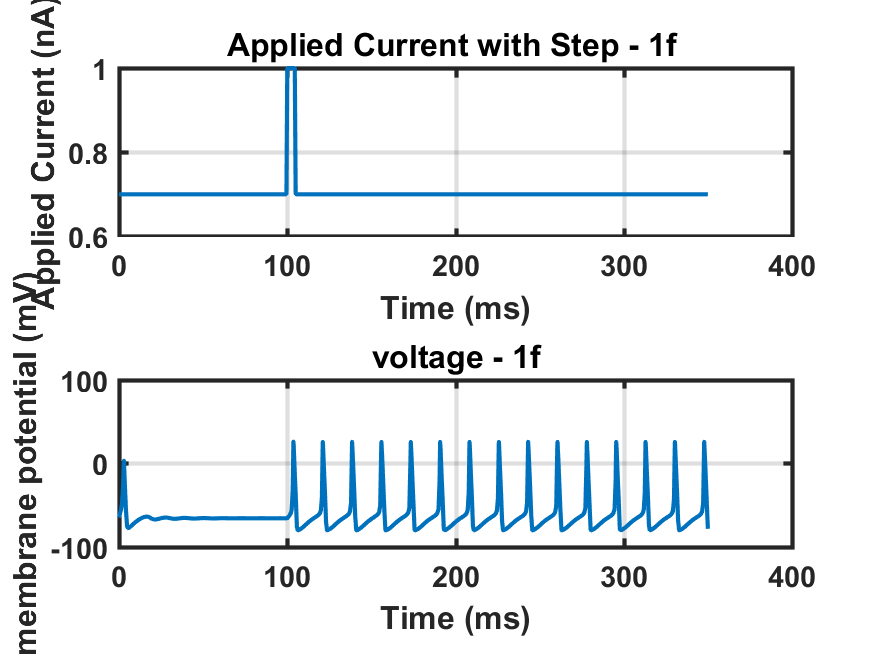
xlabel('Time (ms)')

ylabel('membrane potential (mV)')

title('voltage - 1f')

grid on

saveas(f8, sprintf('1\_f.png'));



Spiking resonance is started just by applying the a spike at 100ms