

# Gopal\_Gupta\_20CH30008\_project\_2.m

```
% Name - Gopal Gupta
% Roll No. - 20CH30008
clc;
clear;
warning off;

moris_lecar_simulation;

hodgkin_huxley_simulation;
function moris_lecar_simulation
% -----Question - 1 (Moris lecar Equation parameter) -----
% Parameter defining
global phi;
global I_ext;
global G_Ca;
global G_K;
global G_l;
global V_Ca;
global V_K;
global V_l;
global V_1;
global V_2;
global V_3;
global V_4;
global V_5;
global V_6;
global C;

% -----Question - 2 (Generating the phase plane plot with nulclines and
equilibrium point)-----
G_Ca = 4.4 ; % mS/cm^2
G_K = 8.0; % mS/cm^2
G_l = 2; % mS/cm^2
V_Ca = 120; % mV
V_K = -84; % mv
V_l = -60; %mV
phi = 0.02; % ms^-1
V_1 = -1.2; %mV
V_2 = 18; %mV
V_3 = 2; %mV
V_4 = 30; %mV
V_5 = 2; %mV
V_6 = 30; %mV
C = 20; % μF/cm^2
I_ext = 0; % μA/cm^2

[V, w] = meshgrid(-80:1.2:40, 0:0.01:1);
dVdt = (1/C)*(I_ext - G_Ca * (0.5*(1+tanh((V-V_1)/V_2))).*(V-V_Ca) - G_K * w .* (V - V_K) - G_l * (V - V_l));
dwdt = phi * (((0.5 * (1+tanh((V-V_3)/V_4)))-w) .* cosh((V-V_3)/V_4));
% plotting the quiver and nullcline
figure(1);
quiver(V, w*100 ,dVdt,dwdt*100,0.9,LineWidth=1);
```

```

hold on;
plot(V(1,:),((I_ext - G_Ca * (0.5*(1+tanh((V(1,:)-V_1)/V_2))).*(V(1,:)-V_Ca)- G_l * (V(1,:)- V_1))./(G_K * (V(1,:)- V_K)))*100,LineWidth=1);
plot(V(1,:),(0.5* (1+tanh((V(1,:)-V_3)/V_4)))*100,LineWidth=1); % plot of null
clines dv/dt=0
%axis equal;
set(gca,'FontSize',20);
axis([-80 40 0 100]);

% finding the equilibrium point by vpasolve and plotting it
syms V w
Vnc1_eqn = (1/C)*(I_ext - G_Ca*(0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K*w*(V-V_K) - G_l*(V-V_1)) == 0;
wnc1_eqn = (0.5*(1+tanh((V-V_3)/V_4)) - w) == 0;
eq_pt_1 = vpasolve([Vnc1_eqn, wnc1_eqn], [V, w]);
V_res = double(eq_pt_1.V);
w_0 = double(eq_pt_1.w);
fprintf('\n----- Question 2 ----- \n');
fprintf('The equilibrium point is located at V_eq = %f mV, w_eq = %f. \n', V_res, w_0);
plot(V_res,w_0*100,'ro');

%% -----Question - 3 (finding the jacobian and eigenvalues) -----
%finding the jacobian and eigenvalues
syms Del_V w_v;
jaco = jacobian([(1/C)*(I_ext - G_Ca * (0.5*(1+tanh((Del_V-V_1)/V_2)))*(Del_V-V_Ca) - G_K * w_v * (Del_V - V_K) - G_l * (Del_V - V_1)),(phi * ((0.5 * (1+tanh((Del_V-V_3)/V_4))-w_v) * cosh((Del_V-V_3)/V_4)))], [Del_V, w_v]);
jaco_eq = double(subs(jaco, {Del_V, w_v}, {V_res, w_0}));
eigenv = eig(jaco_eq);
Det_J = double(det(jaco_eq));
tra_J = double(trace(jaco_eq));
fprintf('\n----- Question 3 ----- \n');
fprintf('The equilibrium point V_eq = %f mV, w_eq = %f is stable because Det(J) = %f > 0 and Trace(J) = %f < 0.\n',..., V_res, w_0,Det_J,tra_J);
fprintf('The eigen values are %f%+fi , %f%+fi \n', real(eigenv(1)), imag(eigenv(1)), ... real(eigenv(2)), imag(eigenv(2)));

%% -----Question - 4 -----
% Solving the Equation using the odefunction

% Options = odeset('RelTol',1e-3,'AbsTol',1e-12, 'refine',5, 'MaxStep', 1);
%% -----Question - 5 (Action potential Generating using MLE) -----
df_func = @moris_lecar_dt;

[t1,Ap1] = ode15s(df_func, [0 300], [0 , 0.014]);
[t2,Ap2] = ode15s(df_func, [0 300], [-12 , 0.014]);
[t3,Ap3] = ode15s(df_func, [0 300], [-24 , 0.014]);
%plotting the trajectories
plot(Ap1(:,1),Ap1(:,2)*100,'b',Ap2(:,1),Ap2(:,2)*100,'b',Ap3(:,1),Ap3(:,2)*100,'b')

```

```

phi = 0.04;
[t1,Ap1] = ode15s(df_func, [0 300], [0 , 0.014]);
[t2,Ap2] = ode15s(df_func, [0 300], [-12 , 0.014]);
[t3,Ap3] = ode15s(df_func, [0 300], [-24 , 0.014]);
%plotting the trajectories
plot(Ap1(:,1),Ap1(:,2)*100,'g',Ap2(:,1),Ap2(:,2)*100,'g',Ap3(:,1),Ap3(:,2)*100,'g')

phi = 0.01;
[t1,Ap1] = ode15s(df_func, [0 300], [0 , 0.014]);
[t2,Ap2] = ode15s(df_func, [0 300], [-12 , 0.014]);
[t3,Ap3] = ode15s(df_func, [0 300], [-24 , 0.014]);
%plotting the trajectories
plot(Ap1(:,1),Ap1(:,2)*100,'m',Ap2(:,1),Ap2(:,2)*100,'m',Ap3(:,1),Ap3(:,2)*100,'m');
legend('', 'Nullcline of V', 'nullcline of w', '', 'Φ = 0.02', '', 'Φ = 0.04', '', 'Φ = 0.01');
xlabel('Voltage(mV)'); ylabel('w * 100'); title('Phase plane plot with nullclines - V and w * 100');
hold off

%% -----Question - 6 (Depolarization Threshold) -----
figure(2);
phi = 0.02;
[V, w] = meshgrid(-70:0.55:40, 0:0.0025:0.5);
dVdt = (1/C)*(I_ext - G_Ca * (0.5*(1+tanh((V-V_1)/V_2))).*(V-V_Ca) - G_K * w .* (V - V_K) - G_L * (V - V_1));
dwdt = phi * (((0.5 * (1+tanh((V-V_3)/V_4)))-w) .* cosh((V-V_3)/V_4));
% plotting the quiver and nullcline
quiver(V, w*100, dVdt, dwdt*100, 0.6, LineWidth=1);
hold on;
plot(V(1,:), ((I_ext - G_Ca * (0.5*(1+tanh((V(1,:)-V_1)/V_2))).*(V(1,:)-V_Ca) - G_L * (V(1,:)-V_1))./(G_K * (V(1,:)-V_K)))*100, LineWidth=1);
plot(V(1,:),(0.5* (1+tanh((V(1,:)-V_3)/V_4)))*100, LineWidth=1);
%axis equal;
set(gca, 'FontSize', 20);
axis([-20 10 0 40]);
plot(V_res, w_0*100, 'ro');
xlabel('Voltage(mV)'); ylabel('w * 100');
title('Phase plane plot with nullclines - V and w * 100');
for V_x = -15.4:0.05:-14.5
[t1,Ap1] = ode15s(df_func, [0 300], [V_x, 0.014]);
plot(Ap1(:,1),Ap1(:,2)*100, LineWidth=0.5);
end
hold off;
figure(3);
V_x = -14.935:0.001:-14.93;
hold on;
for i = 1:length(V_x)
[t1,Ap1] = ode15s(df_func, [0 200], [V_x(i), 0.014]);
plot(t1,Ap1(:,1), LineWidth=0.1);
end
ylabel('Voltage(mV)'); xlabel('Time(ms)');
title('Maximum amplitude of the action potential versus initial value of V');
set(gca, 'FontSize', 20);
hold off;
%axis([0 40 -20 20]);
figure(4);
V_x = -14.9321:0.000001:-14.9320;
hold on;

```

```

for i = 1:length(V_x)
[t1,Ap1] = ode15s(df_func, [0 300], [V_x(i), 0.014]);
plot(V_x(i),max(Ap1(:,1)),'*');
end
ylabel('Maximum amplitude of the action potential(mV)'); xlabel('Initial value of V(mV)');
title('Maximum amplitude of the action potential versus initial value of V');
set(gca,'FontSize',15);
hold off;

ini_V=linspace(-15.2,-14.8,400);
max_V = zeros(1,400);

fprintf(' \n ----- Part 6 ----- \n');
flag=1;
for n = 1:400
[t1,Ap1] = ode15s(df_func,[0 300], [ini_V(n),w_0]);
max_V(n) = max(Ap1(:,1));
if max_V(n) >= 0 && flag==1
fprintf("Threshold is %f mV",ini_V(n));
flag=0;
end
end

%% -----Question - 7 (response for I_ext = 86) -----
I_ext = 86;
[V, w] = meshgrid(-80:1.2:40, 0:0.01:1);
dVdt = (1/C)*(I_ext - G_Ca * (0.5*(1+tanh((V-V_1)/V_2))).*(V-V_Ca) - G_K * w .* (V - V_K) - G_L * (V - V_L));
dwdt = phi * (((0.5 * (1+tanh((V-V_3)/V_4)))-w) .* cosh((V-V_3)/V_4));
% plotting the quiver and nullcline
figure(5);
quiver(V, w*100 ,dVdt,dwdt*100,0.9,LineWidth=1);
hold on;
plot(V(1,:),((I_ext - G_Ca * (0.5*(1+tanh((V(1,:)-V_1)/V_2))).*(V(1,:)-V_Ca)- G_L * (V(1,:)- V_L))./(G_K * (V(1,:)- V_K)))*100,LineWidth=1); % plot of null clines
du/dt=0
plot(V(1,:),(0.5* (1+tanh((V(1,:)-V_3)/V_4)))*100,LineWidth=1); % plot of null clines dv/dt=0
%axis equal;
set(gca,'FontSize',15);
axis([-80 40 0 100]);
% finding the equilibrium point by vpasolve and plotting it
syms V w
Vnc1_eqn = (1/C)*(I_ext - G_Ca*(0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K*w*(V-V_K) - G_L*(V-V_L)) == 0;
wnc1_eqn = (0.5*(1+tanh((V-V_3)/V_4)) - w) == 0;
eq_pt_1 = vpasolve([Vnc1_eqn, wnc1_eqn], [V, w]);
V_res = double(eq_pt_1.V);
w_0 = double(eq_pt_1.w);
fprintf(' \n----- Part 7 ----- \n');
fprintf('The equilibrium point is located at V_eq = %f mV,w_eq = %f \n', V_res, w_0);
plot(V_res,w_0*100,'ro');

%finding the jacobian and eigenvalues
syms Del_V w_v;

```

```

jaco = jacobian([(1/C)*(I_ext - G_Ca * (0.5*(1+tanh((Del_V-V_1)/V_2)))*(Del_V-V_Ca) - G_K * w_v * (Del_V - V_K) - G_l * (Del_V - V_1))), (phi * ((0.5 * (1+tanh((Del_V-V_3)/V_4))) - w_v) * cosh((Del_V-V_3)/V_4))], [Del_V, w_v]);
jaco_eq = double(subs(jaco, {Del_V, w_v}, {V_res, w_0}));
eigenv = eig(jaco_eq);
Det_J = det(jaco_eq);
tra_J = trace(jaco_eq);
fprintf('The equilibrium point is stable because Det(J) = %f > 0 and Trace(J) = %f < 0.\n', Det_J, tra_J);
fprintf('The eigen values are %f%+fi , %f%+fi \n', real(eigenv(1)), imag(eigenv(1)), ...
real(eigenv(2)), imag(eigenv(2)));
[t1,Ap1] = ode15s(df_func, [0 1000], [-60.8554, 0.0149]);
[t2,Ap2] = ode15s(df_func, [0 1000], [V_res, w_0]);
[t3,Ap3] = ode15s(df_func, [0 1000], [-27.9, 0.17]);
plot(Ap1(:,1), Ap1(:,2)*100, 'b', Ap2(:,1), Ap2(:,2)*100, 'r', Ap3(:,1), Ap3(:,2)*100, 'm');
legend('', 'Nullcline of V', 'nullcline of w', '', 'Equilibrium point of I_{ext} = 0 μA/cm^2',...
'Equilibrium point of I_{ext} = 86 μA/cm^2', 'Point with initial values [-27.9, 0.17]');
xlabel('Voltage(mV)'); ylabel('w * 100');
title('Phase plane plot with nullclines - V and w * 100');
hold off;

%% -----Question - 8 (Plotting the Unstable periodic orbit by running backward in time) -----
[V, w] = meshgrid(-80:1.2:40, 0:0.01:1);
dVdt = (1/C)*(I_ext - G_Ca * (0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K * w .* (V - V_K) - G_l * (V - V_1));
dwdt = phi * ((0.5 * (1+tanh((V-V_3)/V_4))) - w) .* cosh((V-V_3)/V_4));
% plotting the quiver and nullcline
figure(6);
quiver(V, w*100, dVdt, dwdt*100, 0.9, LineWidth=1);
hold on;
plot(V(1,:), ((I_ext - G_Ca * (0.5*(1+tanh((V(1,:)-V_1)/V_2)))*(V(1,:)-V_Ca) - G_l * (V(1,:)-V_1))./(G_K * (V(1,:)-V_K)))*100, LineWidth=1); % plot of null clines
du/dt=0
plot(V(1,:),(0.5* (1+tanh((V(1,:)-V_3)/V_4)))*100, LineWidth=1); % plot of null clines dv/dt=0
%axis equal;
set(gca, 'FontSize', 15);
axis([-80 40 0 100]);
% finding the equilibrium point by vpasolve and plotting it
syms V w
Vnc1_eqn = (1/C)*(I_ext - G_Ca*(0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K*w*(V-V_K) - G_l*(V-V_1)) == 0;
wnc1_eqn = (0.5*(1+tanh((V-V_3)/V_4)) - w) == 0;
eq_pt_1 = vpasolve([Vnc1_eqn, wnc1_eqn], [V, w]);

V_res = double(eq_pt_1.V);
w_0 = double(eq_pt_1.w);
plot(V_res, w_0*100, 'ro');

plot(Ap1(:,1), Ap1(:,2)*100, 'b', Ap2(:,1), Ap2(:,2)*100, 'r', Ap3(:,1), Ap3(:,2)*100, 'm');
[t4,Ap4] = ode15s(df_func, [0 -2000], [-27.9, 0.17]);
plot(Ap4(:,1), Ap4(:,2)*100, 'k');
legend('', 'Nullcline of V', 'nullcline of w', '', 'Equilibrium point of I_{ext} = 0 μA/cm^2',...
'Equilibrium point of I_{ext} = 86 μA/cm^2', 'Point with initial values [-27.9, 0.17]');

```

```

'Equilibrium point of I_{ext} = 86 μA/cm^2', 'Point with initial values [-27.9,
0.17]', ...
'UPO in R&E starting at [-27.9, 0.17]');
xlabel('Voltage(mV)'); ylabel('w * 100');
title('Phase plane plot with nullclines - V and w * 100');
hold off;
% -----Question - 9 (Analysing for different external current and plotting firing
rate of AP vs I_ext) -----
fprintf('\n----- Part 9 ----- \n');
for I_ext = [80,86,90]
fprintf('\n----- I_ext = %d μA/cm^2 -----'
--- \n', I_ext);
% finding the equilibrium point by vpasolve and plotting it
syms V w
Vnc1_eqn = (1/C)*(I_ext - G_Ca*(0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K*w*(V-V_K)
- G_l*(V-V_1)) == 0;
wnc1_eqn = (0.5*(1+tanh((V-V_3)/V_4)) - w) == 0;
eq_pt_1 = vpasolve([Vnc1_eqn, wnc1_eqn], [V, w]);

V_eq1 = double(eq_pt_1.V);
w_eq1 = double(eq_pt_1.w);
% Stability Analysis for equilibrium point
fprintf('The equilibrium point is located at V_eq = %f mV, w_eq = %f \n', V_eq1,
w_eq1);

%finding the jacobian and eigenvalues
syms Del_V w_v;
jaco = jacobian([(1/C)*(I_ext - G_Ca * (0.5*(1+tanh((Del_V-V_1)/V_2)))*(Del_V-
V_Ca) - G_K * w_v * (Del_V - V_K) - G_l * (Del_V - V_1))), (phi * ((0.5 *
(1+tanh((Del_V-V_3)/V_4))-w_v) * cosh((Del_V-V_3)/V_4)))], [Del_V, w_v]);
jaco_eq = double(subs(jaco, {Del_V, w_v}, {V_res, w_0}));
eigenv = eig(jaco_eq);
Det_J = det(jaco_eq);
tra_J = trace(jaco_eq);
fprintf('The eigen values are %f%+fi , %f%+fi \n', real(eigenv(1)),
imag(eigenv(1)), ...
real(eigenv(2)), imag(eigenv(2)));
end
ext_current = 80:1:100;
rate_of_firing = zeros(length(ext_current),1);
for i = 1:length(ext_current)
I_ext = ext_current(i);
% finding the equilibrium point by fsolve and plotting it
syms V w
Vnc1_eqn = (1/C)*(I_ext - G_Ca*(0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K*w*(V-V_K)
- G_l*(V-V_1)) == 0;
wnc1_eqn = (0.5*(1+tanh((V-V_3)/V_4)) - w) == 0;
eq_pt_1 = vpasolve([Vnc1_eqn, wnc1_eqn], [V, w]);
V_eq1 = double(eq_pt_1.V);
w_eq1 = double(eq_pt_1.w);

[t3,Ap3] = ode15s(df_func, [0 1000], [V_eq1 + 0.1, w_eq1 + 0.001]);
rate_of_firing(i) = give_frequency(t3,Ap3);
end
figure(7);
plot(ext_current,rate_of_firing)
set(gca,'FontSize',20);

```

```

xlabel('I_{ext} μA/cm^2'); ylabel('Rate of Firing of Action Potential(s^{-1}))');
title('Rate of Firing Action Potentials versus Applied Current');

%% Stable and unstable manifolds (Question 10)
fprintf('\n----- Part 10 ----- \n');
df_func = @moris_lecar_dt;
G_Ca = 4;
phi = 0.0667;
V_3 = 12;
V_4 = 17.4;
I_ext = 30;
figure(8);
[V, w] = meshgrid(-80:1.2:40, 0:0.01:1);
dVdt = (1/C)*(I_ext - G_Ca * (0.5*(1+tanh((V-V_1)/V_2))).*(V-V_Ca) - G_K * w .* (V - V_K) - G_l * (V - V_1));
dwdt = phi * (((0.5 * (1+tanh((V-V_3)/V_4))-w) .* cosh((V-V_3)/V_4));
quiver(V, w*100 ,dVdt,dwdt*100,0.9,LineWidth=1);
hold on;
plot(V(1,:),((I_ext - G_Ca * (0.5*(1+tanh((V(1,:)-V_1)/V_2))).*(V(1,:)-V_Ca)- G_l *(V(1,:)-V_1))./(G_K * (V(1,:)-V_K)))*100,LineWidth=1); % plot of null clines
du/dt=0
plot(V(1,:),(0.5* (1+tanh((V(1,:)-V_3)/V_4)))*100,LineWidth=1); % plot of null clines dv/dt=0
%axis equal;
set(gca,'FontSize',15);
axis([-80 40 -2 100]);
xlabel('Voltage(mV)');ylabel('w * 100');title('Phase plane plot with nullclines - V and w * 100');
% finding the equilibrium points by vpasolve and plotting it
syms V w
Vnc_eqn = (1/C)*(I_ext - G_Ca*(0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K*w*(V-V_K) - G_l*(V-V_1)) == 0;
wnc_eqn = (0.5*(1+tanh((V-V_3)/V_4)) - w) == 0;
eq_pt_1 = vpasolve([Vnc_eqn, wnc_eqn], [V, w],[-40,0]);
eq_pt_2 = vpasolve([Vnc_eqn, wnc_eqn], [V, w],[-20,0.02]);
eq_pt_3 = vpasolve([Vnc_eqn, wnc_eqn], [V, w],[4,0.28]);
V_eq1 = double(eq_pt_1.V);
w_eq1 = double(eq_pt_1.w);
V_eq2 = double(eq_pt_2.V);
w_eq2 = double(eq_pt_2.w);
V_eq3 = double(eq_pt_3.V);
w_eq3 = double(eq_pt_3.w);
plot(V_eq1,w_eq1*100,'ro',V_eq2,w_eq2*100,'bo',V_eq3,w_eq3*100,'go');

%finding the jacobian and eigenvalues
syms Del_V w_v;
jaco = jacobian([(1/C)*(I_ext - G_Ca * (0.5*(1+tanh((Del_V-V_1)/V_2)))*(Del_V-V_Ca) - G_K * w_v * (Del_V - V_K) - G_l * (Del_V - V_1)),(phi * (((0.5 * (1+tanh((Del_V-V_3)/V_4))-w_v) * cosh((Del_V-V_3)/V_4)))], [Del_V, w_v]);
V_eq = [V_eq1, V_eq2, V_eq3];
w_eq = [w_eq1, w_eq2, w_eq3];
Det_J = zeros(3,1);tra_J = zeros(3,1);
eigVectors = 0;
for i =1:3
jaco_eq = double(subs(jaco, {Del_V, w_v}, {V_eq(i), w_eq(i)}));
eigenv = eig(jaco_eq);
Det_J(i) = det(jaco_eq);

```

```

tra_J(i) = trace(jaco_eq);
if and(Det_J(i)>0,tra_J(i)<0)
fprintf('Equilibrium point : %d. (%f, %f) is stable equilibrium point.
\n',i,V_eq(i), w_eq(i))
end
if and(Det_J(i)<0,tra_J(i)<0)
fprintf('Equilibrium point : %d. (%f, %f) is saddle point. \n',i,V_eq(i), w_eq(i))
end
if and(Det_J(i)>0,tra_J(i)>0)
fprintf('Equilibrium point : %d. (%f, %f) is unstable equilibrium point.
\n',i,V_eq(i), w_eq(i))
end
if i == 2
[eigVectors, D] = eig(jaco_eq);
end
fprintf('The corresponding eigen values are %f%+fi, %f%+fi.\n', real(eigenv(1)),
imag(eigenv(1)), ...
real(eigenv(2)), imag(eigenv(2)));
end

%plotting the manifold of saddle point
ev_1 = eigVectors(:,1)';
ev_2 = eigVectors(:,2)';

% Evaluating the manifold
start_1 = [V_eq2,w_eq2] + 1 * ev_1;
start_2 = [V_eq2,w_eq2] + 1 * ev_2;
start_3 = [V_eq2,w_eq2] - 1 * ev_1;
start_4 = [V_eq2,w_eq2] - 1 * ev_2;
[t5,Ap5] = ode15s(df_func, [0 100], start_1);
plot(Ap5(:,1),Ap5(:,2)*100,'k');
[t5,Ap5] = ode15s(df_func, [0 -80], start_2);
plot(Ap5(:,1),Ap5(:,2)*100,'m');
[t5,Ap5] = ode15s(df_func, [0 100], start_3);
plot(Ap5(:,1),Ap5(:,2)*100,'k');
[t5,Ap5] = ode15s(df_func, [0 -30], start_4);
plot(Ap5(:,1),Ap5(:,2)*100,'m');
legend('', 'Nullcline of V', 'Nullcline of w', 'Equilibrium Point 1', 'Equilibrium
Point 2',...
'Equilibrium Point 3', 'Unstable Manifold', 'Stable Manifold');
hold off;

%% Changing the current to see the effect on the equibrium points(Question 11)
fprintf('\n----- Part 11 ----- \n');
cur_ext =
[30,34,38,39,39.1,39.2,39.3,39.4,39.5,39.6,39.7,39.8,39.9,40,41,42,45,50];
freq_1 = zeros(length(cur_ext),1);
for i = 1:length(cur_ext)
I_ext = cur_ext(i);
if and(I_ext>=39,I_ext<42)
figure();
[V, w] = meshgrid(-80:1.2:40, 0:0.01:1);
dVdt = (1/C)*(I_ext - G_Ca * (0.5*(1+tanh((V-V_1)/V_2))).*(V-V_Ca) - G_K * w .* (V
- V_K) - G_L * (V - V_L));
dwdt = phi * (((0.5 * (1+tanh((V-V_3)/V_4)))-w) .* cosh((V-V_3)/V_4));
quiver(V, w*100 ,dVdt,dwdt*100,0.9,LineWidth=1);
hold on;

```

```

plot(V(1,:),((I_ext - G_Ca * (0.5*(1+tanh((V(1,:)-V_1)/V_2)))*(V(1,:)-V_Ca) - G_L * (V(1,:) - V_1))./(G_K * (V(1,:) - V_K)))*100,LineWidth=1);
plot(V(1,:),(0.5* (1+tanh((V(1,:)-V_3)/V_4)))*100,LineWidth=1);
%axis equal;
set(gca,'FontSize',15);
axis([-80 40 -2 100]);
xlabel('Voltage(mV)');ylabel('w * 100');
title(strcat('Phase plane plot with nullclines - V and w * 100 for I_{ext} = ',num2str(I_ext)));
end
% finding the equilibrium points by vpasolve and plotting it
syms V w
Vnc_eqn = (1/C)*(I_ext - G_Ca*(0.5*(1+tanh((V-V_1)/V_2)))*(V-V_Ca) - G_K*w*(V-V_K) - G_L*(V-V_1)) == 0;
wnc_eqn = (0.5*(1+tanh((V-V_3)/V_4)) - w) == 0;
eq_pt_1 = vpasolve([Vnc_eqn, wnc_eqn], [V, w],[-40,0]);
eq_pt_2 = vpasolve([Vnc_eqn, wnc_eqn], [V, w],[-20,0.02]);
eq_pt_3 = vpasolve([Vnc_eqn, wnc_eqn], [V, w],[4,0.28]);
V_eq1 = double(eq_pt_1.V);
w_eq1 = double(eq_pt_1.w);
V_eq2 = double(eq_pt_2.V);
w_eq2 = double(eq_pt_2.w);
V_eq3 = double(eq_pt_3.V);
w_eq3 = double(eq_pt_3.w);
if isequal([V_eq1,w_eq1],[V_eq2,w_eq2])
fprintf('For I_{ext} = %f there is only one distinct equilibrium point. \n',I_ext);
[t,Ap] = ode15s(df_func, [0 2000], [V_eq3 + 0.1,w_eq3 + 0.01]);
if and(I_ext>=39,I_ext<42)
plot(V_eq3, w_eq3*100, 'bo');
plot(Ap(:,1),Ap(:,2)*100);
end
freq_1(i) = give_frequency(t,Ap);
else
fprintf('For I_{ext} = %f, there are three equilibrium points \n',I_ext);
[t,Ap] = ode15s(df_func, [0 2000], [V_eq2 - 0.1,w_eq2 - 0.01]);
freq_1(i) = 0;
if and(I_ext>=39,I_ext<42)
plot(V_eq1,w_eq1*100,'ro',V_eq2,w_eq2*100,'bo',V_eq3,w_eq3*100,'go');
plot(Ap(:,1),Ap(:,2)*100);
end
end
end
hold off;
figure;
plot(cur_ext,freq_1);
xlabel('I_{ext} μA/cm^2');ylabel('Rate of Firing of Action Potential(s^{-1})');
title('Rate of Firing Action Potentials versus Applied Current');
set(gca,'FontSize',17);
end
function hodgkin_huxley_simulation
%% Question 12 (defining the hh model variable)
% Parameter Declaring
global G_K;
global G_Na;
global G_L;
global V_K;
global V_Na;

```

```

global V_L;
global C;
global I_ext;
global cor_fac;

% Parameter defining
C = 1; % μF/cm^2
G_K = 36; % mS/cm^2
G_Na = 120; % mS/cm^2
G_L = 0.3; % mS/cm^2
V_K = -72; % mV
V_Na = 55; % mV
I_ext = 0; % μA/cm^2
cor_fac = 1e-10; %mV

% Accounting for the correction factor such that the system does not get 0/0 case
options = odeset('RelTol',1e-9,'AbsTol',1e-9, 'refine',5, 'MaxStep', 1);
%% Question 13
% Running the model to get the resting potential = -60 mV
V_rs = -60; % mV
alpha_n = -0.01 * (V_rs + cor_fac + 50)/(exp(-(V_rs + cor_fac + 50)/10)-1);
alpha_m = -0.1 * (V_rs + cor_fac + 35)/(exp(-(V_rs + cor_fac + 35)/10)-1);
alpha_h = 0.07 * exp(-(V_rs + 60)/20);

beta_n = 0.125 * exp(-(V_rs + 60)/80);
beta_m = 4 * exp(-(V_rs + 60)/18);
beta_h = 1/(exp(-(V_rs + 30)/10)+1);

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

E_leak = V_rs - (1/G_L) * (I_ext- G_K * n_inf^4 * (V_rs - V_K) -G_Na * m_inf^3 * h_inf * (V_rs - V_Na));
fprintf('----- Part 13 ----- \n');
fprintf('E_L = %f mV \n', double(E_leak));

V_L = E_leak;
% Generating Action potential
I_ext = 10;
hh_model = @hodgkin_huxley;
Ap_ini = [-60, n_inf, m_inf, h_inf];
[t1, Ap1] = ode15s(hh_model, [0 100], Ap_ini, options);
figure;
subplot(2,1,1);
plot(t1,Ap1(:,1));
xlabel('Time (ms)');ylabel('Voltage (mV)');title('Generating Action potential for HH model');
set(gca,'FontSize',17);
subplot(2,1,2);
plot(t1,Ap1(:,2),'r',t1,Ap1(:,3),'m',t1,Ap1(:,4),'k');
xlabel('Time (ms)');ylabel('Probability of open channel');
title('Getting Variable probability versus time');
legend('n-act (K)', 'm-act (Na)', 'h-Inact (Na)');
set(gca,'FontSize',17);

%% Question 14

```

```
% Checking the Stability for I_ext = 0
fprintf('\n----- Part 14 ----- \n');
fprintf('Checking the Stability for I_ext = 0 μA/cm^2\n');
checking_stability(0);

% Current threshold
I_ext = 0;
implse_curr = linspace(0,15,100);
max_V = zeros(length(implse_curr),1);
for i = 1:length(implse_curr)
Ap_ini = [-60+(implse_curr(i)/C), n_inf, m_inf, h_inf];
[t1, Ap1] = ode15s(hh_model, [0 100], Ap_ini, options);
max_V(i) = max(Ap1(:,1));
end
threshold = (min(max_V(:)) + max(max_V(:)))/2;
curr_index = 500;
for i = 1:100
if max_V(i) >= threshold && i < curr_index
curr_index = i;
end
end
fprintf('\n The Threshold of current is %f μA/cm^2.\n',implse_curr(curr_index));
figure;
plot(implse_curr,max_V);
xlabel('Impulse Current (μA/cm^2)');ylabel('Peak Voltage (mV)');
title('Threshold of the HH model for brief current pulses');
set(gca,'FontSize',17)

%% Observering the behaviour for I_ext = 8 to 12 (Question 15)
fprintf('\n----- Part 15 ----- \n');
for i = 8:12
fprintf("\n I_ext = %d μA/cm^2 \n", i);
checking_stability(i);
end

I_ext = 8;
Ap_ini = [-55, 0.4, 0, 0.4];
[t1, Ap1] = ode15s(hh_model, [0 100], Ap_ini, options);
figure;
subplot(2,1,1);
plot(t1,Ap1(:,1));
xlabel('Time (ms)');ylabel('Volatge (mV)');
title('For I_{ext} = 8 μA/cm^2 with Ap_{ini} = [-55, 0.4, 0, 0.4]');
set(gca,'FontSize',15);
grid on;
subplot(2,1,2);
hold on;
plot(t1,Ap1(:,2),'r',t1,Ap1(:,3),'m',t1,Ap1(:,4),'k');
xlabel('Time (ms)');ylabel('Probability of open channel');
title('Gatting Variable probablity versus time');
legend('n-act (K)', 'm-act (Na)', 'h-Inact (Na)');
set(gca,'FontSize',15);
grid on;
hold off;

I_ext = 9;
Ap_ini = [-54, 0.4, 0, 0.4];
```

```
[t1, Ap1] = ode15s(hh_model, [0 100], Ap_ini, options);
figure;
subplot(2,1,1);
plot(t1,Ap1(:,1));
xlabel('Time (ms)');ylabel('Volatge (mV)');
title('For I_{ext} = 9 μA/cm^2 with Ap_{ini} = [-54, 0.4, 0, 0.4]');
set(gca,'FontSize',15);
grid on;
subplot(2,1,2);
hold on;
plot(t1,Ap1(:,2),'r',t1,Ap1(:,3),'m',t1,Ap1(:,4),'k');
xlabel('Time (ms)');ylabel('Probability of open channel');
title('Gatting Variable probablity versus time');
legend('n-act (K)', 'm-act (Na)', 'h-Inact (Na)');
set(gca,'FontSize',15);
grid on;
hold off;

I_ext = 10;
Ap_ini = [-54, 0.4, 0, 0.4];
[t1, Ap1] = ode15s(hh_model, [0 100], Ap_ini, options);
figure;
subplot(2,1,1);
plot(t1,Ap1(:,1));
xlabel('Time (ms)');ylabel('Volatge (mV)');
title('For I_{ext} = 10 μA/cm^2 with Ap_{ini} = [-54, 0.4, 0, 0.4]');
set(gca,'FontSize',15);
grid on;
subplot(2,1,2);
hold on;
plot(t1,Ap1(:,2),'r',t1,Ap1(:,3),'m',t1,Ap1(:,4),'k');
xlabel('Time (ms)');ylabel('Probability of open channel');
title('Gatting Variable probablity versus time');
legend('n-act (K)', 'm-act (Na)', 'h-Inact (Na)');
set(gca,'FontSize',15);
grid on;
hold off;

I_ext = 11;
Ap_ini = [-54, 0.4, 0, 0.4];
[t1, Ap1] = ode15s(hh_model, [0 100], Ap_ini, options);
figure;
subplot(2,1,1);
plot(t1,Ap1(:,1));
xlabel('Time (ms)');ylabel('Volatge (mV)');
title('For I_{ext} = 11 μA/cm^2 with Ap_{ini} = [-54, 0.4, 0, 0.4]');
set(gca,'FontSize',15);
grid on;
subplot(2,1,2);
hold on;
plot(t1,Ap1(:,2),'r',t1,Ap1(:,3),'m',t1,Ap1(:,4),'k');
xlabel('Time (ms)');ylabel('Probability of open channel');
title('Gating Variable probablity versus time');
legend('n-act (K)', 'm-act (Na)', 'h-Inact (Na)');
set(gca,'FontSize',15);
grid on;
hold off;
```

```

%% For the V-n reduced system (Question 16)
hh_model_n = @hh_reduced_n;
I_ext = 10;
Ap_ini = [-60, n_inf];
[t1, Ap1] = ode15s(hh_model_n, [0 100], Ap_ini, options);
figure;
subplot(2,1,1);
plot(t1,Ap1(:,1));
xlabel('Time (ms)'); ylabel('Voltage (mV)');
title('Generating Action potential for HH V-n Reduced model');
set(gca,'FontSize',15);
m_in = zeros(length(Ap1(:,1)),1);
for i = 1:length(Ap1(:,1))
V_rs = Ap1(i,1);
alpha_m = -0.1 * (V_rs + cor_fac + 35)/(exp(-(V_rs + cor_fac + 35)/10)-1);
beta_m = 4 * exp(-(V_rs + 60)/18);
m_in(i) = alpha_m/(alpha_m + beta_m);
end
subplot(2,1,2);
plot(t1,Ap1(:,2), 'r', t1, m_in, 'm', t1, (1-Ap1(:,2)), 'k');
xlabel('Time (ms)'); ylabel('Probability of open channel');
title('Gating Variable probability versus time');
legend('n-act (K)', 'm-act (Na)', 'h-Inact (Na)');
set(gca,'FontSize',15);

% Checking for current pulse
I_ext = 0;
implse_curr = linspace(0,15,5);
figure;
xlabel('Time (ms)'); ylabel('Voltage (mV)');
title('Simulating the HH V-n reduced model for brief current pulse');
hold on;
for i = 1:length(implse_curr)
Ap_ini = [-60+(implse_curr(i)/C), n_inf];
[t1, Ap1] = ode15s(hh_model_n, [0 20], Ap_ini, options);
plot(t1,Ap1(:,1));
end
legend(strcat('implse curr (\mu A/cm^2) = ', num2str(implse_curr(1))), strcat('implse curr (\mu A/cm^2)= ', num2str(implse_curr(2))), ...
strcat('implse curr (\mu A/cm^2)= ', num2str(implse_curr(3))), strcat('implse curr (\mu A/cm^2)= ', num2str(implse_curr(4))), ...
strcat('implse curr (\mu A/cm^2)= ', num2str(implse_curr(5))));
set(gca,'FontSize',15);
hold off;

%% Anode break excitation (Question 17)
I_ext = -3;
Ap_ini_1 = [-60, n_inf, m_inf, h_inf];
[t1, Ap1] = ode15s(hh_model, [0 20], Ap_ini_1, options);
Ap_ini_2 = [Ap1(end,1), Ap1(end,2), Ap1(end,3), Ap1(end,4)];
n_anode = Ap1(end,2);
h_anode = Ap1(end,4);

I_ext = 0;
[t2, Ap2] = ode15s(hh_model, [20 100], Ap_ini_2, options);
tot_t = [t1; t2];

```

```

tot_Ap = [Ap1; Ap2];
figure;
subplot(2,1,1);
plot(tot_t,tot_Ap(:,1));
xlabel('Time (ms)'); ylabel('Voltage (mV)');
title('Anode break Excitation for HH model');
set(gca,'FontSize',17);
subplot(2,1,2);
plot(tot_t,tot_Ap(:,2),'r',tot_t,tot_Ap(:,3),'m',tot_t,tot_Ap(:,4),'k');
xlabel('Time (ms)'); ylabel('Probability of open channel');
title('Gating Variable probability versus time');
legend('n-act (K)', 'm-act (Na)', 'h-Inact (Na)');
set(gca,'FontSize',17);

%% For Anode break excitation in V-m reduced hh model(Question 18)
fprintf('\n----- Part 18 ----- \n');
V_rs = -60;
alpha_n = @(V) -0.01 * (V + cor_fac + 50)/(exp(-(V + cor_fac + 50)/10)-1);
alpha_m = @(V) -0.1 * (V + cor_fac + 35)/(exp(-(V + cor_fac + 35)/10)-1);
alpha_h = @(V) 0.07 * exp(-(V + 60)/20);

beta_n = @(V) 0.125 * exp(-(V + 60)/80);
beta_m = @(V) 4 * exp(-(V + 60)/18);
beta_h = @(V) 1/(exp(-(V + 30)/10)+1);

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

I_ext = -3;
Ap_ini_1 = [-60, m_inf];
[t1, Ap1] = ode15s(@(t,Ap)hh_reduced_m(t,Ap,n_inf,h_inf), [0 20], Ap_ini_1,
options);
V_nc_1 = @(V) (((I_ext- G_K * n_inf^4 * (V - V_K) - G_L * (V - V_L))/(G_Na * h_inf * (V - V_Na)))^(1/3));
m_nc = @(V) alpha_m(V)/(alpha_m(V) + beta_m(V));
figure;
hold on
fplot(@(V) V_nc_1(V), [-80 60]);
fplot(@(V) m_nc(V), [-80 60]);
Ap_ini_2 = [Ap1(end,1), Ap1(end,2)];
n_inf1 = n_anode;
h_inf1 = h_anode;
I_ext = 0;
[t2, Ap2] = ode15s(@(t,Ap)hh_reduced_m(t,Ap,n_inf1,h_inf1), [20 100], Ap_ini_2,
options);
V_nc_1 = @(V) (((I_ext- G_K * n_inf1^4 * (V - V_K) - G_L * (V - V_L))/(G_Na * h_inf1 * (V - V_Na)))^(1/3));
fplot(@(V) V_nc_1(V), [-80 60]);

tot_Ap = [Ap1; Ap2];
plot(tot_Ap(:,1),tot_Ap(:,2))
xlabel('Voltage (mV)'); ylabel('m');
title('Phase plane plot for anode break in V-m reduced model');
set(gca,'FontSize',15);
axis([-80 60 0 1.1]);
grid on;

```

```
% Equilibrium points in the both cases
syms V m
I_ext = -3;
V_nc_eq1 = (1/C) * (I_ext- G_K * n_inf^4 * (V - V_K) -G_Na * m^3 * h_inf * (V - V_Na) - G_L * (V - V_L)) == 0;
m_nc_eq1 = alpha_m * (1-m) - beta_m * m == 0;

eq_pt1 = vpasolve([V_nc_eq1, m_nc_eq1],[V, m]);
V_eq1 = double(eq_pt1.V);
m_eq1 = double(eq_pt1.m);
fprintf('The equilibrium point for case 1 is V_eq = %f and m_eq = %f.\n',V_eq1,m_eq1);

I_ext = 0;
V_nc_eq2 = (1/C) * (I_ext- G_K * n_inf1^4 * (V - V_K) -G_Na * m^3 * h_inf1 * (V - V_Na) - G_L * (V - V_L)) == 0;

eq_pt2 = vpasolve([V_nc_eq2, m_nc_eq1],[V, m]);
V_eq2 = double(eq_pt2.V);
m_eq2 = double(eq_pt2.m);
fprintf('The equilibrium point for case 2 is V_eq = %f and m_eq = %f.\n',V_eq2,m_eq2);

plot(V_eq1,m_eq1,'ro');
plot(V_eq2,m_eq2,'bo')
legend('V - nullcline for I_{ext} = -3 μA/cm^2','m - nullcline',...
'V - nullcline for I_{ext} = 0 μA/cm^2','Trajectory Anode break excitation',...
'Equilibrium point for I_{ext} = -3 μA/cm^2','Equilibrium point for I_{ext} = 0 μA/cm^2');
legend('Location','northwest')
end

function dAp_n = hh_reduced_m(t,Ap,n_inf,h_inf)
global G_K;
global G_Na;
global G_L;
global V_K;
global V_Na;
global V_L;
global C;
global I_ext;
global cor_fac;

V = Ap(1);
m = Ap(2);

alpha_m = -0.1 * (V + cor_fac + 35)/(exp(-(V + cor_fac + 35)/10)-1);
beta_m = 4 * exp(-(V + 60)/18);

dV = (1/C) * (I_ext- G_K * n_inf^4 * (V - V_K) -G_Na * m^3 * h_inf * (V - V_Na) - G_L * (V - V_L));
dm = alpha_m * (1-m) - beta_m * m;

dAp_n = [dV; dm];
end
```

```

function dAp_n = hh_reduced_n(t,Ap)
global G_K;
global G_Na;
global G_L;
global V_K;
global V_Na;
global V_L;
global C;
global I_ext;
global cor_fac;

V = Ap(1);
n = Ap(2);

alpha_n = -0.01 * (V + cor_fac + 50)/(exp(-(V + cor_fac + 50)/10)-1);
alpha_m = -0.1 * (V + cor_fac + 35)/(exp(-(V + cor_fac + 35)/10)-1);

beta_n = 0.125 * exp(-(V + 60)/80);
beta_m = 4 * exp(-(V + 60)/18);

m_inf = alpha_m/(alpha_m + beta_m);

dV = (1/C) * (I_ext- G_K * n^4 * (V - V_K) -G_Na * m_inf^3 * (1-n) * (V - V_Na) -
G_L * (V - V_L));
dn = alpha_n * (1-n) - beta_n * n;

dAp_n = [dV; dn];
end

function dAp = hodgkin_huxley(t,Ap)
global G_K;
global G_Na;
global G_L;
global V_K;
global V_Na;
global V_L;
global C;
global I_ext;
global cor_fac;

V = Ap(1);
n = Ap(2);
m = Ap(3);
h = Ap(4);

alpha_n = -0.01 * (V + cor_fac + 50)/(exp(-(V + cor_fac + 50)/10)-1);
alpha_m = -0.1 * (V + cor_fac + 35)/(exp(-(V + cor_fac + 35)/10)-1);
alpha_h = 0.07 * exp(-(V + 60)/20);

beta_n = 0.125 * exp(-(V + 60)/80);
beta_m = 4 * exp(-(V + 60)/18);
beta_h = 1/(exp(-(V + 30)/10)+1);

dV = (1/C) * (I_ext- G_K * n^4 * (V - V_K) -G_Na * m^3 * h * (V - V_Na) - G_L * (V -
V_L));
dn = alpha_n * (1-n) - beta_n * n;
dm = alpha_m * (1-m) - beta_m * m;
dh = alpha_h * (1-h) - beta_h * h;
```

```

dh = alpha_h * (1-h) - beta_h * h;

dAp = [dV; dn; dm; dh];
end

function checking_stability(I_ext)
global G_K;
global G_Na;
global G_L;
global V_K;
global V_Na;
global V_L;
global C;
global cor_fac;

syms V n m h
alpha_n = -0.01 * (V + cor_fac + 50)/(exp(-(V + cor_fac + 50)/10)-1);
alpha_m = -0.1 * (V + cor_fac + 35)/(exp(-(V + cor_fac + 35)/10)-1);
alpha_h = 0.07 * exp(-(V + 60)/20);

beta_n = 0.125 * exp(-(V + 60)/80);
beta_m = 4 * exp(-(V + 60)/18);
beta_h = 1/(exp(-(V + 30)/10)+1);

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

V_nc = (1/C) * (I_ext- G_K * n^4 * (V - V_K) -G_Na * m^3 * h * (V - V_Na) - G_L * (V - V_L)) == 0;
n_nc = alpha_n * (1-n) - beta_n * n == 0;
m_nc = alpha_m * (1-m) - beta_m * m == 0;
h_nc = alpha_h * (1-h) - beta_h * h == 0;

eq_pt = vpasolve([V_nc, n_nc, m_nc, h_nc], [V, n, m ,h]);
V_eq1 = double(eq_pt.V);
n_eq1 = double(eq_pt.n);
m_eq1 = double(eq_pt.m);
h_eq1 = double(eq_pt.h);

for i = 1:length(eq_pt)
fprintf('The Equilibrium points are V_eq = %f mV, n_eq = %f, m_eq = %f. h_eq = %f \n',...
eq_pt(i).V, eq_pt(i).n, eq_pt(i).m, eq_pt(i).h);
end

dV = (1/C) * (I_ext- G_K * n^4 * (V - V_K) -G_Na * m^3 * h * (V - V_Na) - G_L * (V - V_L));
dn = alpha_n * (1-n) - beta_n * n;
dm = alpha_m * (1-m) - beta_m * m;
dh = alpha_h * (1-h) - beta_h * h;

jaco = jacobian([dV, dn, dm, dh],[V, n, m, h]);
jaco_eq = double(subs(jaco, {V, n, m, h}, {V_eq1, n_eq1, m_eq1, h_eq1}));
eigenV = eig(jaco_eq);
fprintf('The corresponding EigenValues are:\n');
disp(eigenV)

```

```
check=0;
for i=1:4
if real(eigenV(i)) < 0
check = check+1;
else
check = check-1;
end
end
if check == 4
fprintf('Equilibrium point is stable equilibrium point. \n')
elseif check == -4
fprintf('Equilibrium point is unstable equilibrium point. \n')
else
fprintf('Cannot say anything about the Equilibrium point (Need to plot in 4
dimensions). \n')
end
end

function freq = give_frequency(t,Ap)
s = size(Ap);
s = s(1);
Ap_V = Ap(:,1);
spike = 0;
for i = 1:s
if Ap_V(i) > 0
spike = 1;
end
end
if spike == 0
freq = 0;
return
end

% Checking for the negative membrane voltage
while Ap_V(s) < 0
s = s-1;
end
% Recording the first positive membrane voltage
while Ap_V(s) > 0
s = s-1;
end
t1 = t(s);
% Checking again for the negative membrane voltage
while Ap_V(s) < 0
s = s-1;
end
% Recording the second positive membrane voltage
while Ap_V(s) > 0
s = s-1;
end
t2 = t(s);
freq = 1000/(t1-t2);
end

function dxdt = moris_lecar_dt(t,x)
dxdt = zeros(2,1);
```

```
global phi;
global I_ext;
global G_Ca;
global G_K;
global G_l;
global V_Ca;
global V_K;
global V_l;
global V_1;
global V_2;
global V_3;
global V_4;
global C;
m_inf = 0.5*(1+tanh((x(1)-V_1)/V_2));
w_inf = 0.5* (1+tanh((x(1)-V_3)/V_4));
tau = 1/(cosh((x(1)-V_3)/V_4));
dxdt(1) = (1/C)*(I_ext - G_Ca * m_inf *(x(1)-V_Ca) - G_K * x(2) * (x(1) - V_K) -
G_l * (x(1) - V_l));
dxdt(2) = (phi * (w_inf - x(2)))/tau;
end
```