

Computational Neuroscience

Project 2 Submission

Group No 6

Siddharth S Jha	14EE30022
Adarsh Mukesh	13BT30031
Pratik Chatse	16MM61R07
Ambati Ashok Kumar	

Part 1: Morris Lecar Model

Question 1:

The following values and corresponding units were chosen.

```
gcal=4.4; % mS/ cm^2
gk=8.0; % mS/ cm^2
gl=2; % mS/ cm^2
vca=120; % mV
vk=-84; % mV
vl=-60; % mV
phi=0.02; %ms^-1
v1=-1.2; % mV
v2=18; % mV
v3=2; % mV
v4=30; % mV
v5=2; % mV
v6=30; % mV
c=20; % uF/cm^2
Iext=0; % uA/cm^2
```

If we want conductance in the form of $\mu\text{S}/\text{cm}^2$, we would have to change the voltage units to be in Volts instead of millivolts. Hence the system of units becomes milliseconds, microamp/ cm^2 , volts, microfarad/ cm^2 and microsiemens/ cm^2 .

No the solution is not unique because we can change the units of voltage and hence corresponding current, capacitance etc. units to the forms, maybe centivolts instead of millivolts and thus scaling down both current and capacitance. So there would never be a single solution of units.

Question 2:

$$V_{eq} = -60.8553$$

$$W_{eq} = 0.01492$$

The equilibrium points are calculated by substituting the value of W calculated by setting $dW/dt=0$ in the $dV/dt=0$ equation and solving the equation for V only.

The following code would deal with the same.

```
syms Vnull;
syms wn;
null1 = (1/c) * (gcal * (0.5 * (1 + tanh((Vnull - v1)/v2))) * (vca - Vnull) +
gk * (0.5 * (1 + tanh((Vnull - v3)/v4))) * (vk - Vnull) + gl * (v1 - Vnull) + Iext);
Veq = solve(null1, Vnull);
Weq = 0.5 * (1 + tanh((Veq - v3)/v4));
```

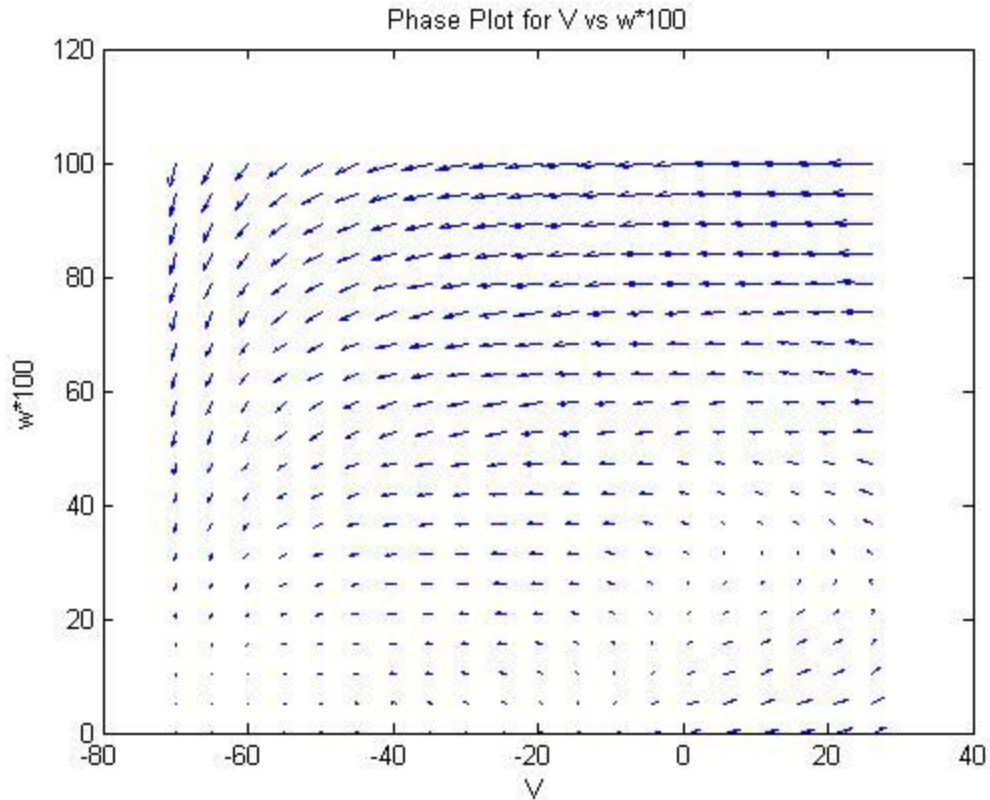
This can also be done by finding the intersection of nullclines using `interx` command.

The quiver Plot was obtained by scaling the w variable by 100. The following code snippet give the quiver plot showing the flow of V and w in phase plane for $I_{ext}=0$

```
iter1=20;
mf=100;
[V0, W0]=meshgrid(-70:((100+1)/iter1):30, 0:(1/(iter1-1)):1);
U=zeros(iter1,iter1);
Vq=zeros(iter1,iter1);

for i=1:iter1*iter1
    U(i)=phi*(0.5*(1+tanh((V0(i)-v3)/v4))-W0(i))*cosh((V0(i)-v3)/v4);
    %dW/dt
    Vq(i)=(Iext - gcal*0.5*(1 + tanh((V0(i) - v1)/v2))*(V0(i)-vca) -
    gk*W0(i)*(V0(i)-vk)-gl*(V0(i)-v1))/c; %dV/dt

end
figure(4)
quiver(V0,W0*mf,Vq,U*mf);
title('Phase Plot for V vs w');
end
```



Question 3:

The evaluated Jacobian is

```
[ -0.1004,  -9.2578]
[ 0.00008, -0.08250]
```

And the eigenvalues calculated using the eig() function of MATLAB are:

```
(- 0.0915 + 0.0258*i)
(- 0.0915 - 0.0258*i)
```

are the eigenvalues obtained for the equilibrium point.

The following code snippet deals with the evaluation of eigenvalues and the jacobian.

```
jac = jacobian([(1/c)*(gcal*(0.5*(1+tanh((Vnull-v1)/v2)))*(vca-Vnull)
+ gk*wn*(vk-Vnull) + gl*(v1-Vnull)+Iext),
phi*((0.5*(1+tanh((Vnull-v3)/v4)))-wn)*cosh((Vnull-v3)/v4)], [Vnull,wn
]);
```

```
jaceq = subs(jac, {sym('Vnull'), sym('wn')}, {Veq, Weq})
eigenv = eig(jaceq);
eigen_stab(eigenv(1),eigenv(2))
```

We've written the function `eigen_stab` that determines the stability of the evaluated eigenvalue. It is found that the eigenvalue is stable.

Question 4:

These values are reasonable as we are dealing mostly with units and values that are precise up to 4 places of decimal (or single values as MATLAB likes to call them) and hence the Absolute tolerance of 10^{-6} is much more than sufficient to estimate the solutions to the differential equations. The relative tolerance of 10^{-3} or 0.1% is also acceptable as we are generating approximate nullclines. If we changed the unit of voltage to kV, the absolute tolerance would be affected badly as we'd require a precision of a great order. For example the -60.8553 mV value would then be read as -0.0000608553 kV and thus 6 more decimal places are required to get the same precision and hence the absolute tolerance would have to be decreased by at least a factor of 10^6 . On the other hand relative tolerance won't be much affected by units

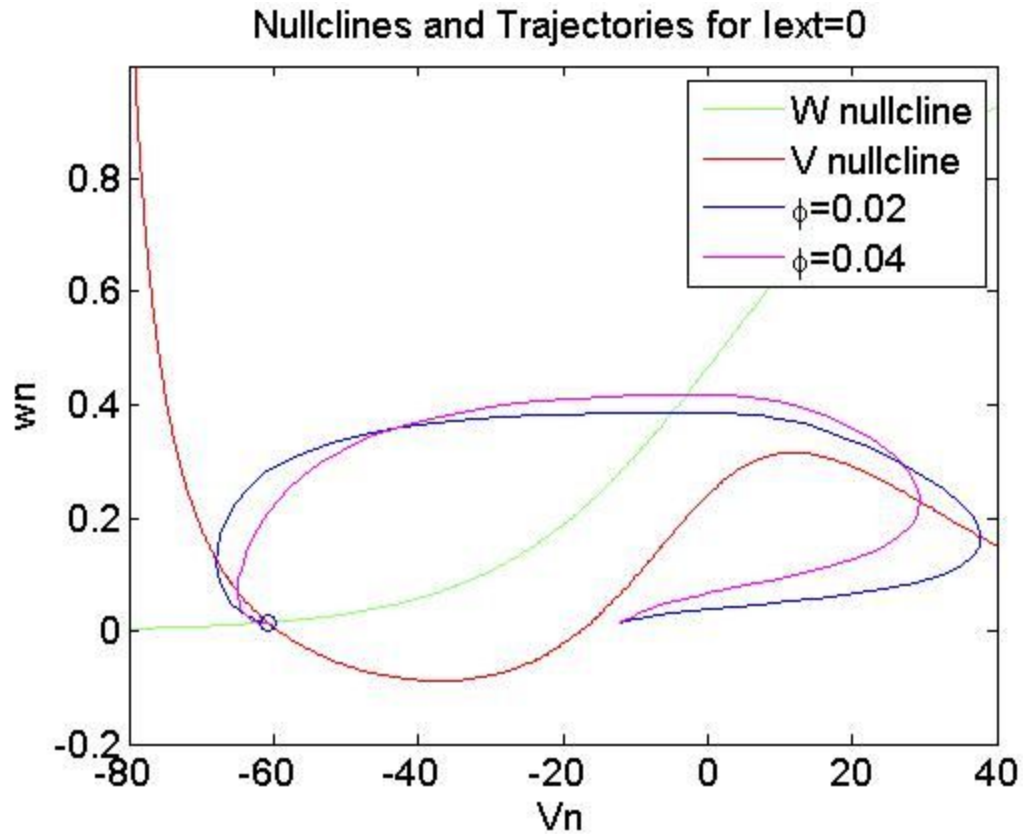
The values can be set in the equation as:

```
Options = odeset('RelTol',1e-3,'AbsTol',1e-12, 'refine',5, 'MaxStep',
1);
```

Question 5:

The Action potential can be generated by calculating the values of the differential equations for a time span, and plotting the values of the state variables in the same plot as the nullclines.

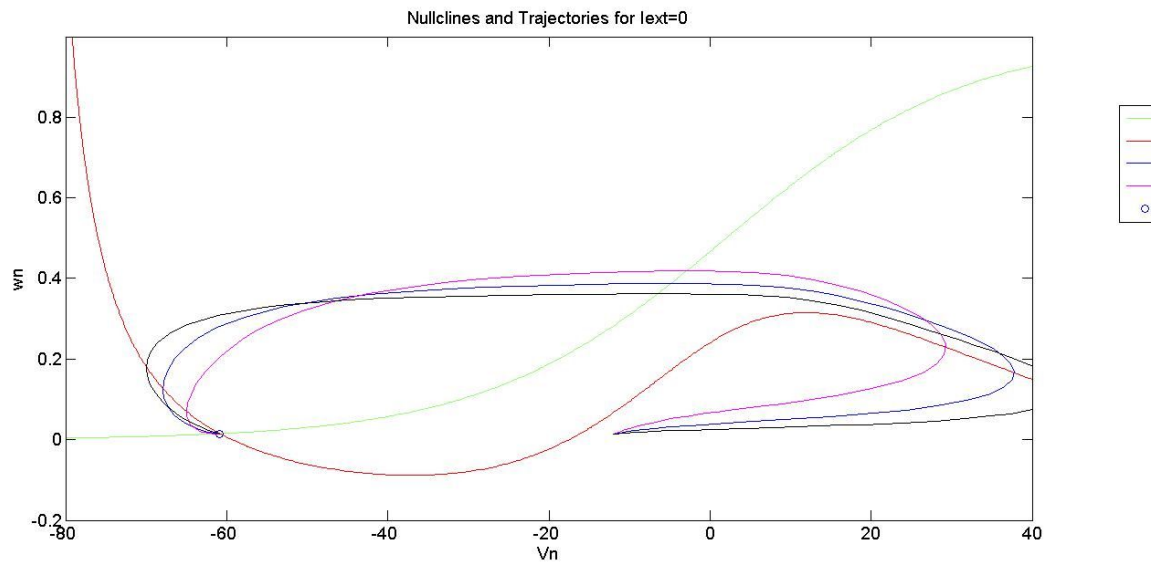
The following plot was obtained for the different values of ϕ , namely 0.02 and 0.04.



The starting point in this case was $(-12, 0.014)$

The nullclines are in red and green and the equilibrium point has been marked with a circle.

When the corresponding plot is run for $\phi=0.01$, the result is the black curve that moves out of the 40mV limit I set on the plot.



Thus we see that with increasing the value of ϕ , the onset of action potential actually becomes faster and the max positive voltage decreases, which is logical since in a way we are decreasing the time constant of w (potassium) activation, hence the drop is faster.

Question 7:

We thus need to increase the value of I_{ext} and run the simulation once again. As expected the V nullcline rises, but there still is just a single equilibrium point. The point is stable once again and the evaluated value comes out to be:

$V_{\text{eq1}} =$

-27.9524

$W_{\text{eq1}} =$

0.1195

And the evaluated Jacobian turns out to be:

$J_{\text{aceq1}} =$

[0.00897, -22.4190]
[0.00022, -0.03082]

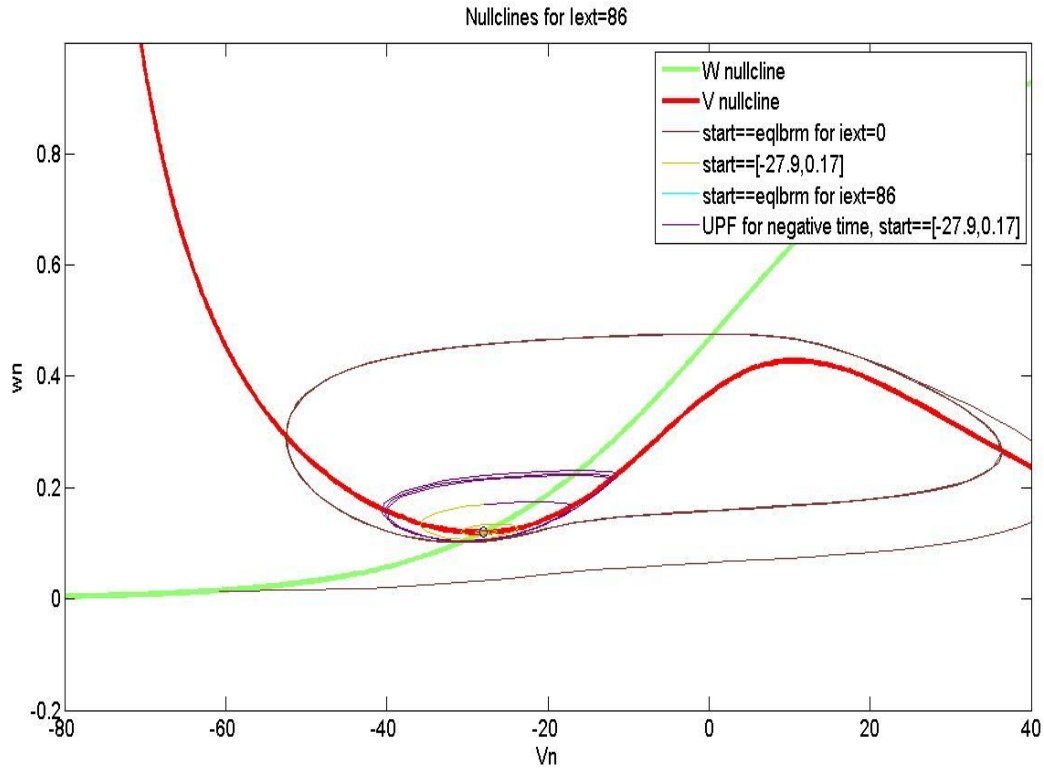
With the eigenvalues as:

$\text{eigenv1} =$

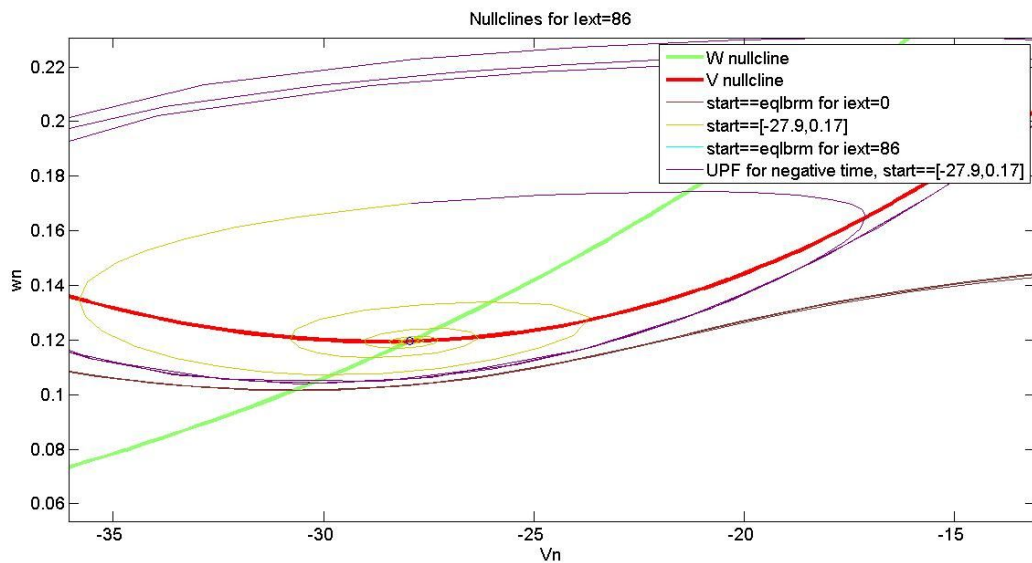
- 0.01092 + 0.06673*i
- 0.01092 - 0.06673*i

Which shows that the point is stable.

The trajectories are once again calculated by solving the differential equation for different initial conditions.



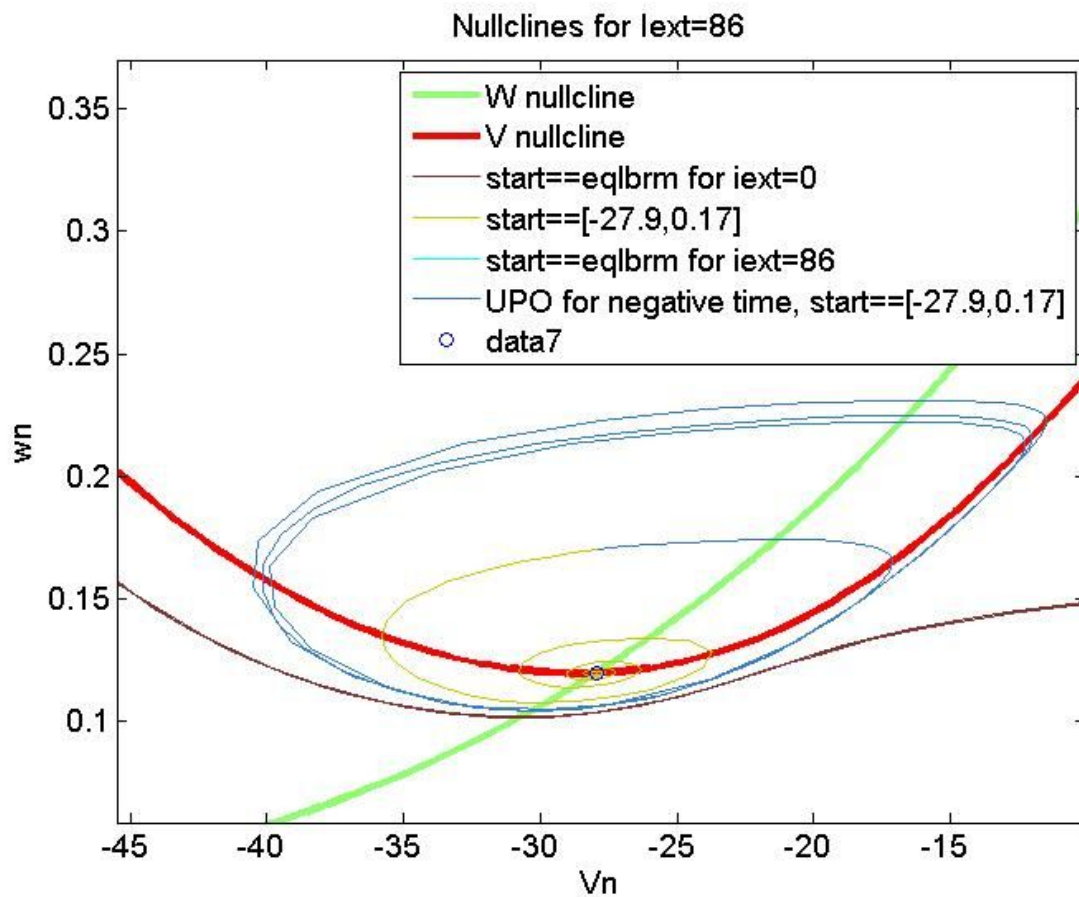
Here the thick green and red lines are the nullclines. The start point when set to the value of the equilibrium for $\text{text}=0$ turns out to force the trajectory into a limit cycle. When the starting point is close, the curve spirals into the equilibrium point which is observed in the dark yellow trajectory (Although it is not very clear in the figure, hence here's a zoomed in view near the equilibrium point).



The first case when the initial condition is set as the equilibrium point for 0 uA gives us the action potentials. This method can be used in patch clamp recording to generate action potentials. The second case can help us to judge the stability of the system at a given point.

Question 8:

The differential equation is solved in negative time for the same point that converged into the equilibrium point i.e. $(-27.9, 0.015)$ and the trajectory came out to be an unstable periodic orbit, which is evident from the bluish-green line emerging from the point $-27.9, 0.015$.



Now any point outside of UPO would end up on a limit cycle and any point inside the UPO will converge to the stable point in the positive time period.

The following code snippet accomplishes all of this.

```
figure(2)
phi=0.02
```



```

Iext=86.0; %New case with Iext 86uA/cm^2
myfun1 = @(Vn,wn) (1/c)*(gcal*(0.5*(1+tanh((Vn-v1)/v2)))*(vca-Vn) +
gk*wn*(vk-Vn) + gl*(v1-Vn)+Iext);
myfun2 = @(Vn,wn)
phi*((0.5*(1+tanh((Vn-v3)/v4)))-wn)/(1/cosh((Vn-v3)/v4));

set(gca, 'fontsize', 14);
%plotting nullclines again
a3=ezplot(@(Vn,wn) myfun2(Vn,wn), [-80 40 -0.2 1])
set(a3,'Linewidth', 3);
hold on
a4=ezplot(@(Vn,wn) myfun1(Vn,wn), [-80 60 -0.2 1])
set(a4,'color',[1 0 0],'Linewidth', 3);

title('Nullclines for Iext=86')

syms Vnull1;
syms wn1;
%evaluating new equilibrium point
null2 =(1/c)*(gcal*0.5*(1+tanh((Vnull1-v1)/v2))*(vca-Vnull1) +
gk*0.5*(1+tanh((Vnull1-v3)/v4))*(vk-Vnull1) + gl*(v1-Vnull1)+Iext);

Veq1 = solve(null2,Vnull1)
Weq1 = 0.5*(1+tanh((Veq1-v3)/v4))

[tp3,Sp3]=ode15s(func, [0 500], [-60.8,0.0149]);
[tp4,Sp4]=ode15s(func, [0 500], [-27.9, 0.17]);
[tp5,Sp5]=ode15s(func, [0 500], [-27.9524,0.1195]); %eq1brm point
manually added
[tp6,Sp6]=ode15s(func, [0 -500], [-27.9, 0.17]);

p3=plot(Sp3(:,1), Sp3(:,2));
set(p3,'color',[0.5 0.2 0.2]);
hold on
p4=plot(Sp4(:,1), Sp4(:,2));
set(p4,'color',[0.8 0.8 0]);
hold on
p5=plot(Sp5(:,1), Sp5(:,2));
set(p5,'color',[0 1 1]);
hold on
p6=plot(Sp6(:,1), Sp6(:,2),'y');
set(p6,'color',[0.2 0.5 0.8]);

```

```

legend('W nullcline','V nullcline','start==eqnbrm for
iext=0','start==[-27.9,0.17]','start==eqnbrm for iext=86','UPO for
negative time, start==[-27.9,0.17]');
plot(Veq1,Weq1,'o');

%evaluating jacobian and eigenvalues
jac1 =
jacobian([(1/c)*(gcal*(0.5*(1+tanh((Vnull1-v1)/v2)))*(vca-Vnull1) +
gk*wn1*(vk-Vnull1) + gl*(v1-Vnull1)+Iext),
phi*((0.5*(1+tanh((Vnull1-v3)/v4)))-wn1)*cosh((Vnull1-v3)/(v4))],[Vnu
ll1,wn1]);
jaceq1 = subs(jac1, {sym('Vnull1'), sym('wn1')}, {Veq1, Weq1})
eigenv1 = eig(jaceq1)
eigen_stab(eigenv1(1),eigenv1(2));

```

Question 10:

For these constants, there are 3 equilibrium points and they are characterized by their eigenvalues as being stable spiral, saddle and stable.

The eigenvalue

p1 =

- 0.3104 + 0.2827*i

,

p2 =

- 0.3104- 0.2827*i

is stable and converging spiral

The eigenvalue

p1 =

0.07467

,

p2 =

-1.9012
is a saddle node

The eigenvalue

p1 =

-0.06465

,

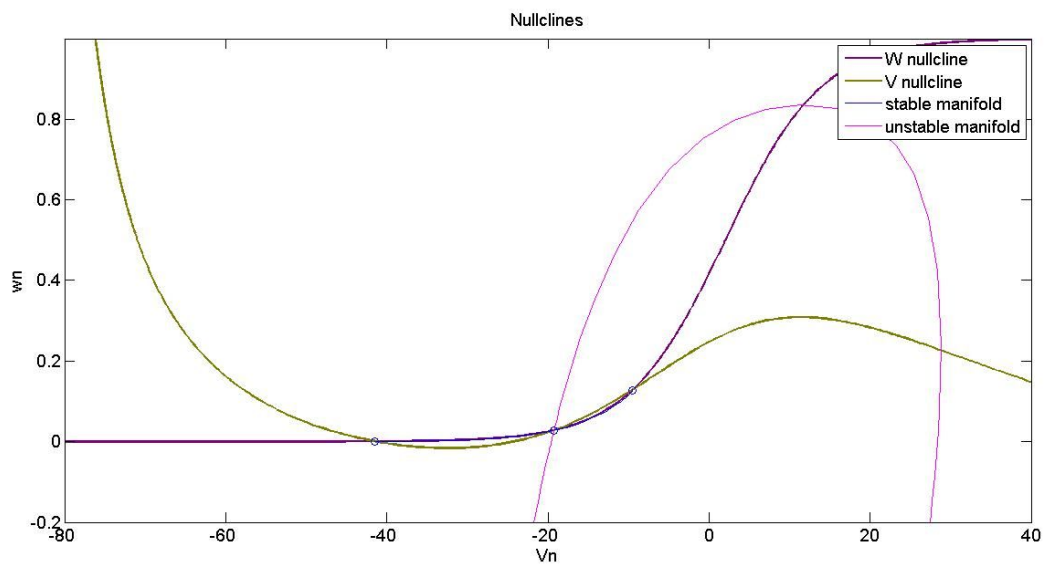
p2 =

-12.5101

is stable

For the manifolds, we need to evaluate the path of points very close to the stable point and run simulations both in positive and negative time and plot the corresponding trajectories.

Here is the plot corresponding to this part.



The blue lines that coincide with the nullclines end up at the stable nodes and the unstable manifolds don't do so (magenta).

Here is the code snippet

```

figure(3)
myfun1 = @(Vn,wn) (1/c)*(gcal*(0.5*(1+tanh((Vn-v1)/v2)))*(vca-Vn) +
gk*wn*(vk-Vn) + gl*(vl-Vn)+Iext);
myfun2 = @(Vn,wn)
phi*((0.5*(1+tanh((Vn-v3)/v4)))-wn)/(1/cosh((Vn-v3)/v4));
set(gca, 'fontsize', 14);

%plotting nullclines again
a5=ezplot(@(Vn,wn) myfun2(Vn,wn), [-80 40 -0.2 1]);
set(a5,'Linewidth',2,'color',[0.5 0 0.5]);
hold on
a6=ezplot(@(Vn,wn) myfun1(Vn,wn), [-80 60 -0.2 1]);
set(a6,'Linewidth',2,'color',[0.5 0.5 0])
title('Nullclines,Manifolds for Iext=30')

%evaluating manifolds
[tp7,Sp7]=ode15s(func, [0 1000], [-19.242, 0.0281]);
[tp8,Sp8]=ode15s(func, [0 -1000], [-19.242, 0.0281]);
[tp9,Sp9]=ode15s(func, [0 1000], [-19.244, 0.0283]);
[tp10,Sp10]=ode15s(func, [0 -1000], [-19.244, 0.0283]);
plot(Sp7(:,1), Sp7(:,2),'b');
hold on
plot(Sp8(:,1), Sp8(:,2),'m');
hold on
plot(Sp9(:,1), Sp9(:,2),'b');
hold on
plot(Sp10(:,1), Sp10(:,2),'m');
legend('W nullcline','V nullcline','stable manifold','unstable
manifold');
plot(Veq2_p1,Weq2_p1,'o')
plot(Veq2_p2,Weq2_p2,'o')
plot(Veq2_p3,Weq2_p3,'o')
hold off

```

Please note that these code snippets aren't independent and require value from the whole code to run. Please run the final project2.m file to view the morris lecar plots.

Part 2: Hodgkin Huxley Model

Question 12:

Following is the code snippet for Hodgkin Huxley

```

global E_Na E_K E_l T_celcius T Cm Q

E_Na=55; %mV
E_K=-72; %mV
E_l=-49; %mV
T_celcius=6.3;
T=273+6.3;%kelvin
Cm=1; %microFarad/cm^2;
Q=3;

global G_Na G_K G_l
G_Na=120; %milliSimens/cm^2;
G_K=36; %milliSimens/cm^2;
G_l=0.3; %milliSimens/cm^2;
I_ext=[0 10 0]; %uA/cm^2

t_tot=50;
curr_start=5;
curr_dur=40; % increase this to generate multiple action potentials
curr_stop=curr_start+curr_dur;

V=-60; %mV
m=0;
h=0.6;
n=0.3;
dynamic_variables=[V m h n];

init_states=[V m h n];
time=0;

[t,vars]=ode15s(@(t,vars) deriv_hh(t,vars,I_ext(1)), [0
curr_start],init_states);
dynamic_variables=[dynamic_variables; vars];
time=[time;t];
[t,vars]=ode15s(@(t,vars) deriv_hh(t,vars,I_ext(2)), [curr_start
curr_stop],dynamic_variables(end,:));
dynamic_variables=[dynamic_variables; vars];
time=[time;t];
[t,vars]=ode15s(@(t,vars) deriv_hh(t,vars,I_ext(3)), [curr_stop
50],dynamic_variables(end,:));
dynamic_variables=[dynamic_variables; vars];
time=[time;t];

```

```

figure
subplot(2,1,1)
plot(time,dynamic_variables(:,1));
hold on
plot((curr_start:0.01:curr_stop),I_ext(2),'r','LineWidth',5);
grid on
xlabel('Time ms')
ylabel('membrane voltage mV')
title('action potential at 10uA')

subplot(2,1,2)
hold on
plot(time,dynamic_variables(:,2),'b');
plot(time,dynamic_variables(:,3),'r');
plot(time,dynamic_variables(:,4),'g');
set(gca,'TickDir','Out');
legend('m','h','n');
grid on
xlabel('time ms')
ylabel('probabilities')

```

Following is the code snippet of the differential equation file of the HH Model:

```

function [ variables ] = deriv_hh(t,states,I_ext)
global G_Na G_K G_l E_Na E_K E_l Q T Cm T_celcius
V=states(1);
m=states(2);
h=states(3);
n=states(4);

g_Na=G_Na*m^3*h;

```

```

I_Na=g_Na*(V-E_Na);

g_K=G_K*n^4;
I_K=g_K*(V-E_K);

g_l=G_l;
I_l=g_l*(V-E_l);

I_ion=I_Na+I_K+I_l;

phi=Q*((T_celcius-6.3)/10);

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

alpha_h=0.07*phi*exp(-(V+60)/20);
beta_h=phi/(exp(-(V+30)/10)+1);

alpha_n=-(0.01*phi*(V+50))/(exp(-(V+50)/10)-1);
beta_n=0.125*phi*exp(-(V+60)/80);

D_m=alpha_m*(1-m)-beta_m*m;
D_h=alpha_h*(1-h)-beta_h*h;
D_n=alpha_n*(1-n)-beta_n*n;
D_V=I_ext-I_ion/Cm;

variables=[D_V;D_m;D_h;D_n];

return

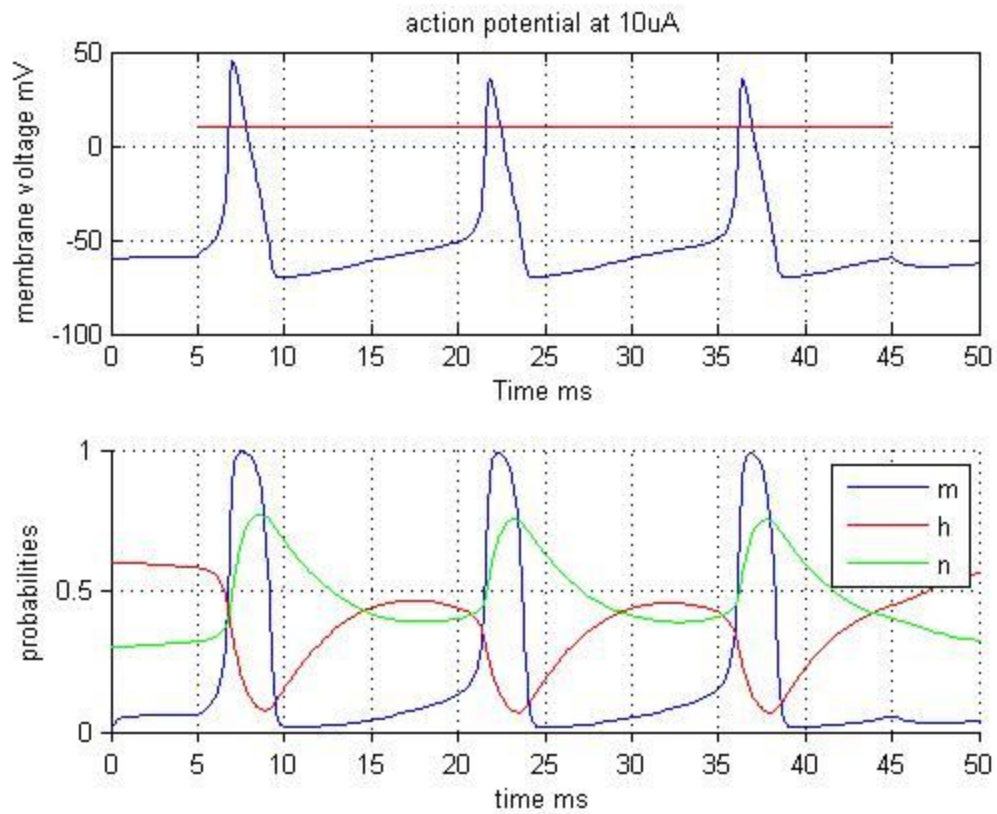
```

To take care for the 0/0 condition, limit at that voltage can be computed using L'Hopital's rule.

Question 13:

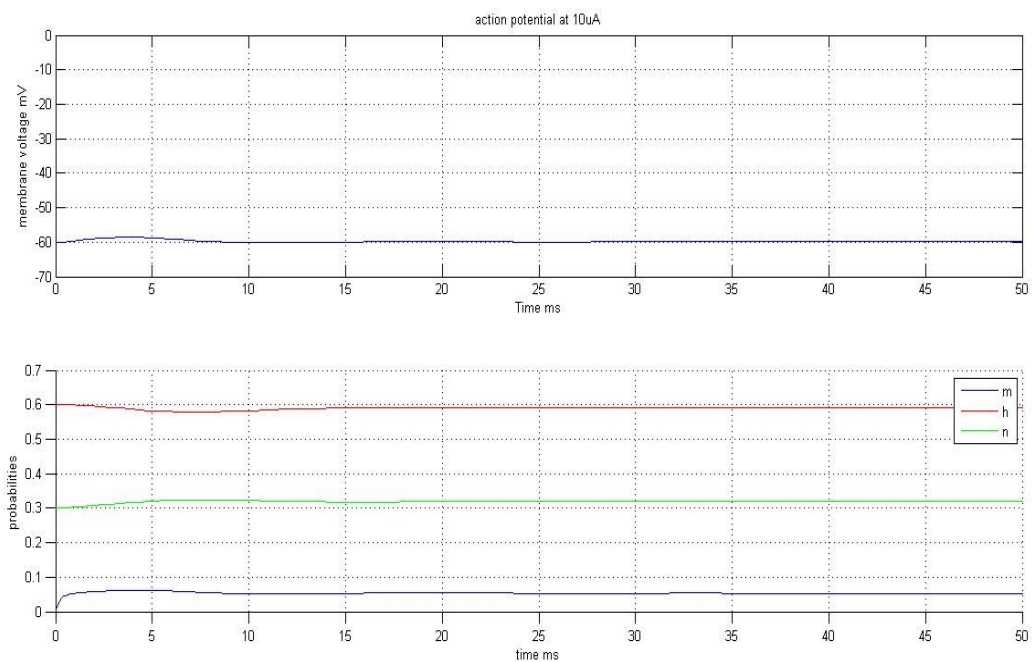
To find EI at which the resting potential is -60mV, we fix dm/dt , dn/dt and dh/dt to 0 and v at -60mV. Then we find the corresponding values of m , h and n . We fix these values into the dv/dt equation and get the value of EI = -49mV.

Fixing these values and a current of 10uA, we get the following plots.

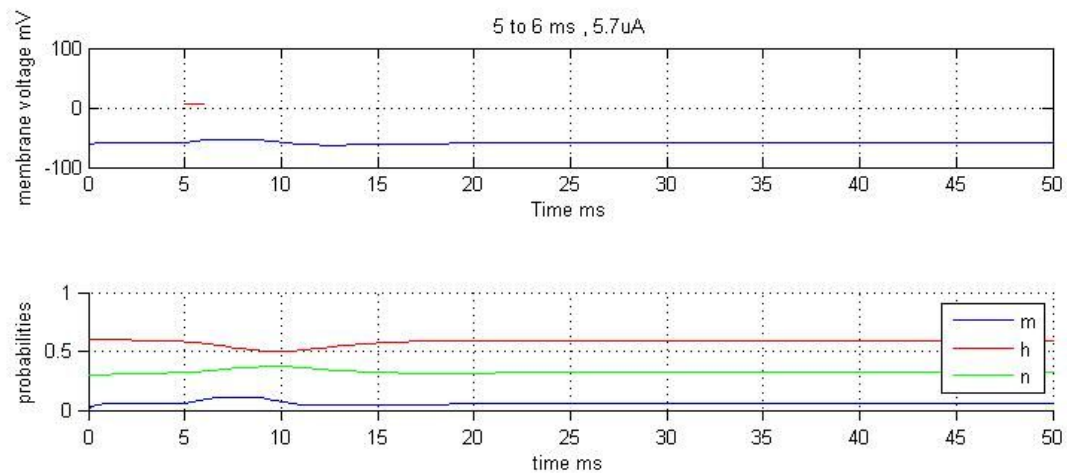


Question 14:

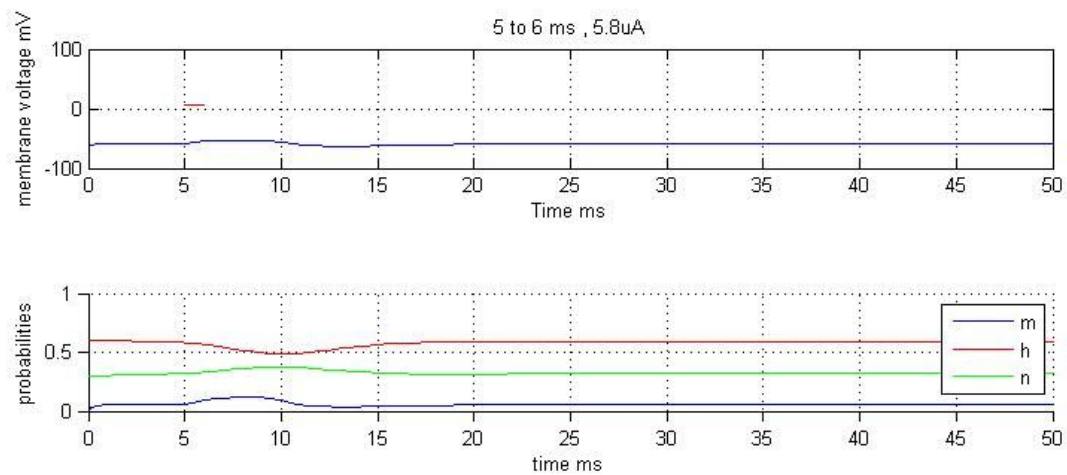
The following shows the stability at $I_{ext} = 0$



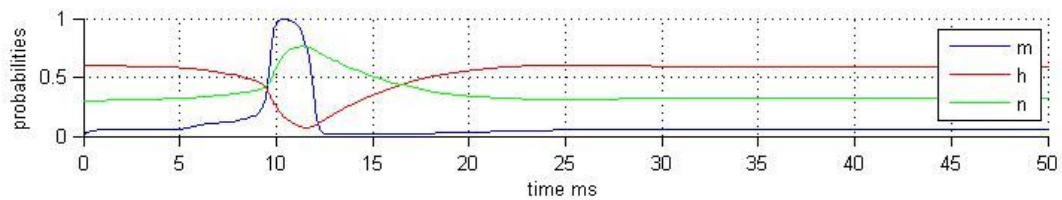
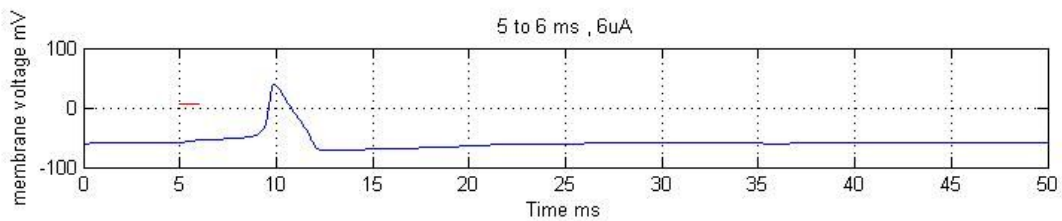
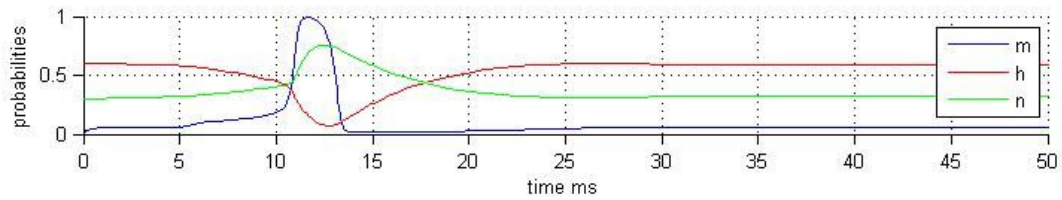
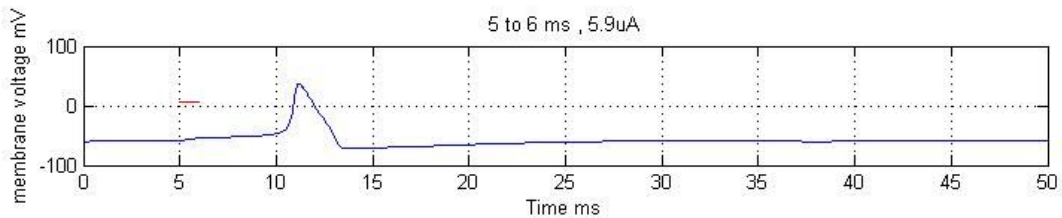
Now we apply current pulse of various amplitudes for 1ms.



At 5.7uA, no action potential is generated. Same is the case for 5.8uA.



At 5.9uA and above, an action potential behaviour is seen.



Question 15:

We found out the equilibrium points from 8uA to 12uA as following:

-55.3054	0.0905	0.4288	0.3914
-54.9055	0.0946	0.4149	0.3978
-54.5278	0.0986	0.4019	0.4038
-54.1696	0.1025	0.3897	0.4095
-53.8290	0.1064	0.3783	0.4149

We use the following code:

```
syms V m h n
```

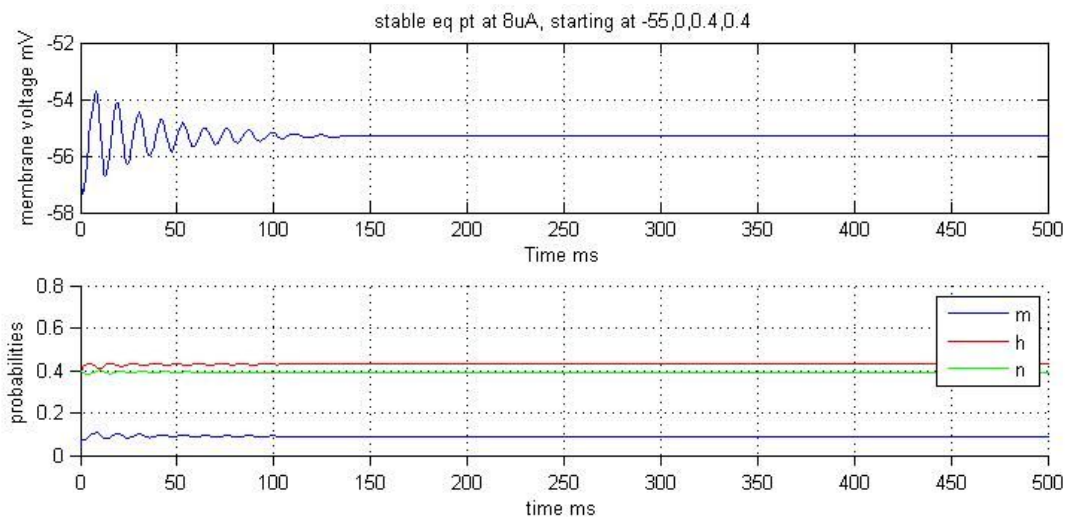
```
% alpha_m=-1*(0.1*(V+35))/(exp(-(V+35)/10)-1);
```

```

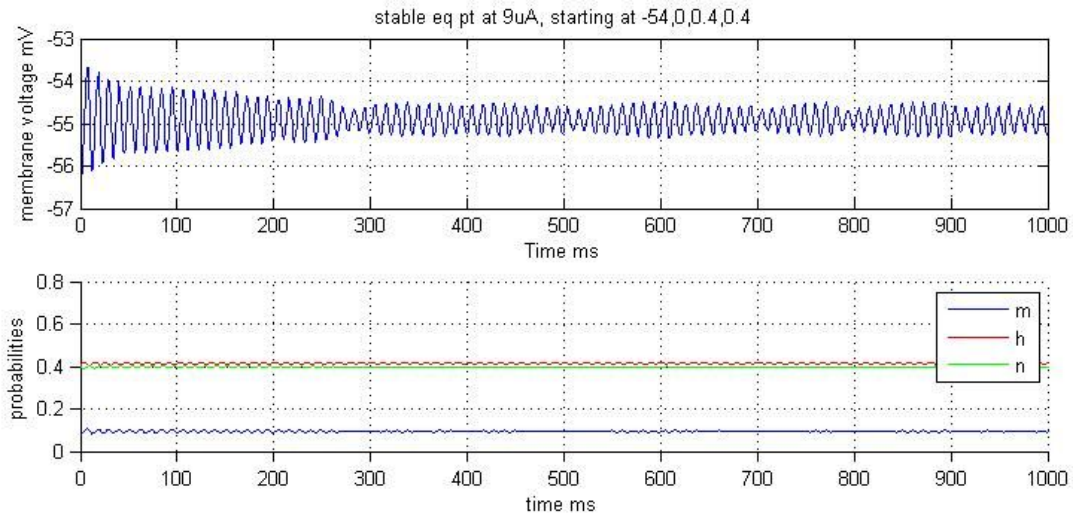
% beta_m=4*exp(-(V+60)/18);
%
% alpha_h=0.07*exp(-(V+60)/20);
% beta_h=1/(exp(-(V+30)/10)+1);
%
% alpha_n=-(0.01*(V+50))/(exp(-(V+50)/10)-1);
% beta_n=0.125*exp(-(V+60)/80);
eq_pts=[];
for i=8:12
eqn1=i-120*m^3*h*(V-55)-36*n^4*(V+72)-0.3*(V+49)==0;
eqn2=(-1*(0.1*(V+35))/(exp(-(V+35)/10)-1))*(1-m)-(4*exp(-(V+60)/18))*
m==0;
eqn3=(0.07*exp(-(V+60)/20))*(1-h)-(1/(exp(-(V+30)/10)+1))*h==0;
eqn4=(-(0.01*(V+50))/(exp(-(V+50)/10)-1))*(1-n)-(0.125*exp(-(V+60)/80
))*n==0;
S=solve(eqn1,eqn2,eqn3,eqn4);
new=[S.V S.m S.h S.n];
eq_pts=[eq_pts; new];
end

```

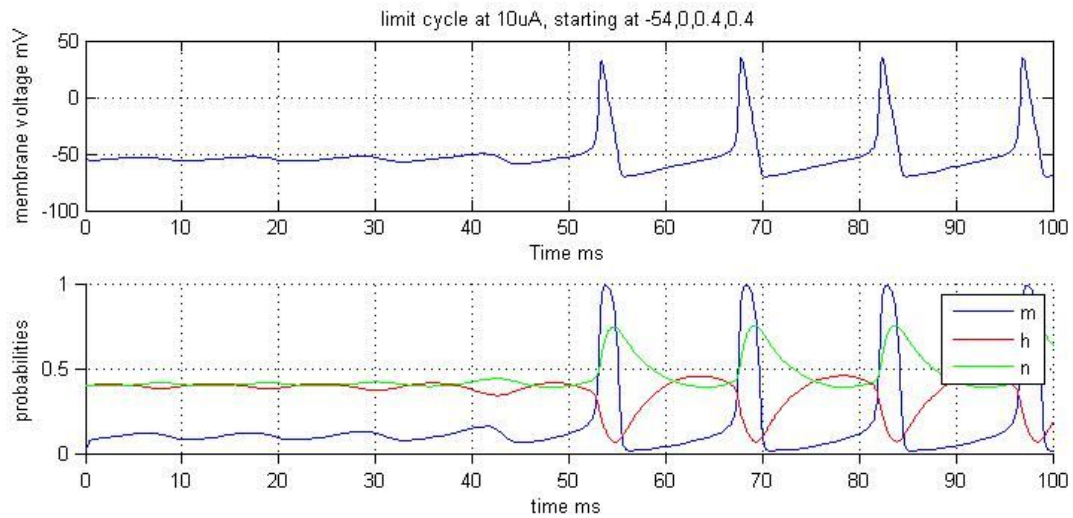
Now we run the model at points near these values for respective currents and find the nature of these equilibrium pts.



We see that at 8uA, the equilibrium point is stable.

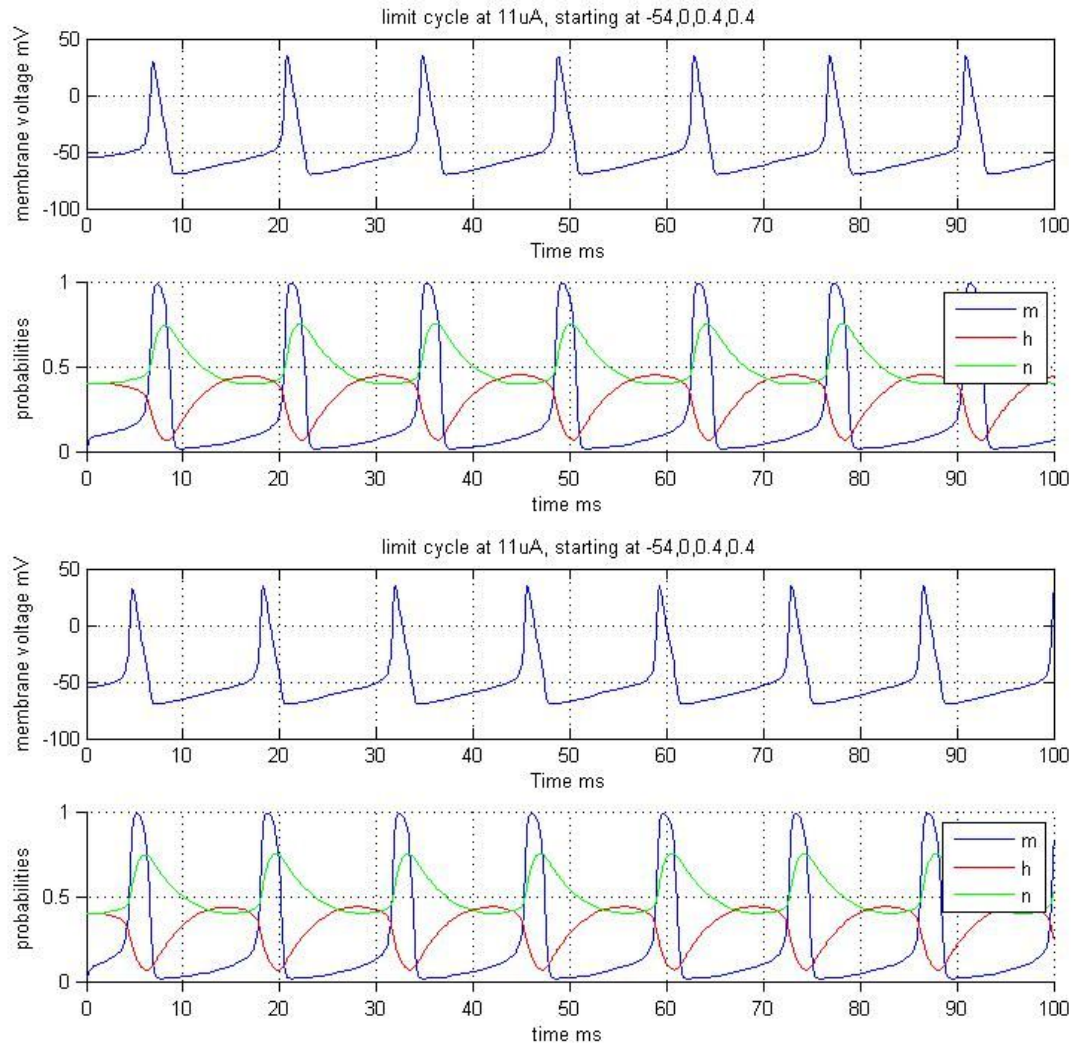


At 9uA, the equilibrium point is stable but takes a longer time to decay



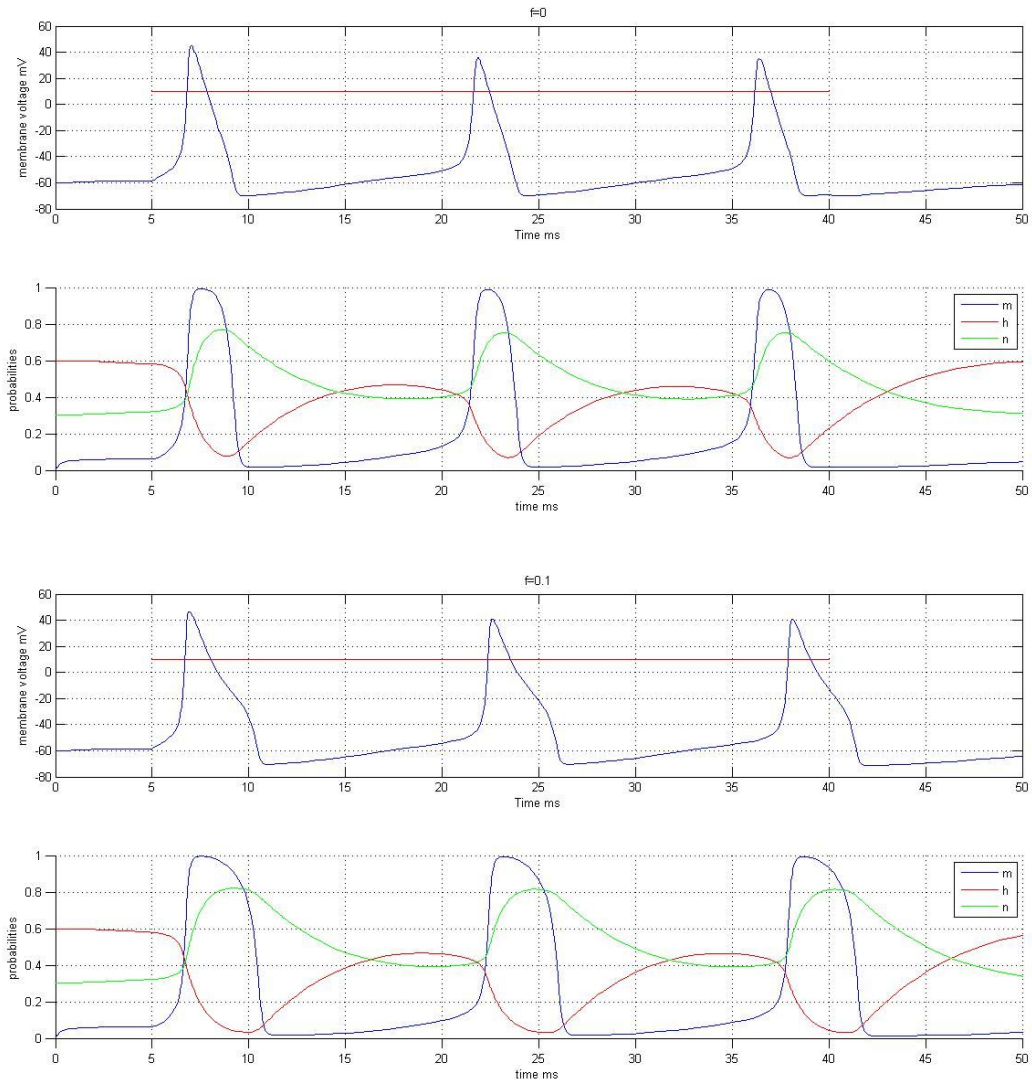
At 10uA, we get a limit cycle after some oscillations.

At 11 and 12uA, the equilibrium points become unstable and we get full limit cycle.

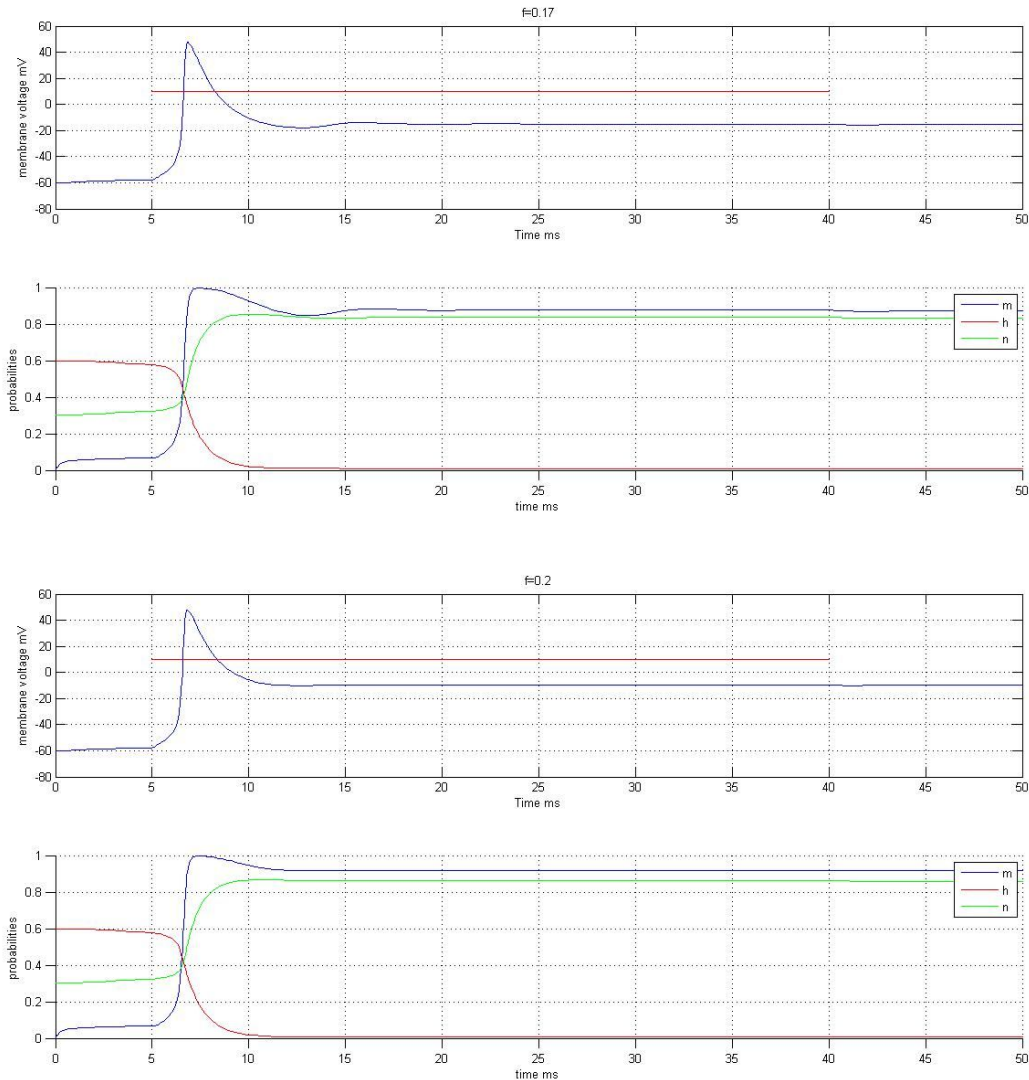


Question 16:

In case of myotonia, we find that for $f=0$, and $f=0.1$, action potentials are generated.



But for $f=0.17$ and 0.2 , action potentials are not generated.



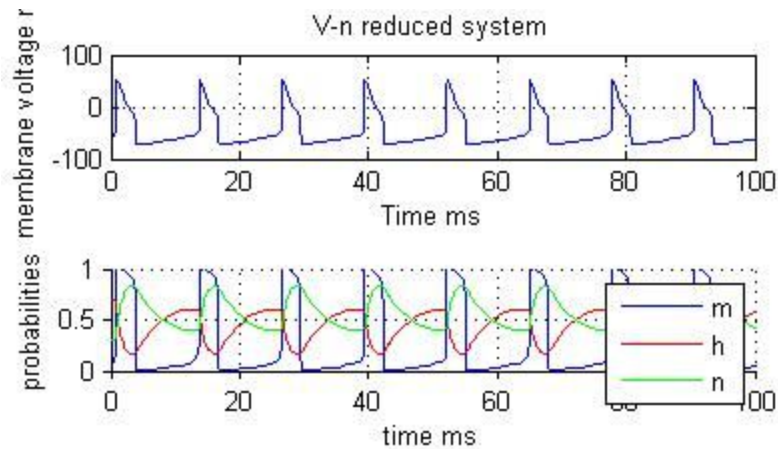
We can see that some fraction of the sodium channels replace their h variable with f and lose their inactivating behaviour. The inactivating behaviour is necessary for the absolute refractory period which brings down the cell to ground state and makes it ready for the next action potential. It can be seen from the above graph that the cell is not coming back to its ground state and hence even though it is getting a stimulus, it is not firing any action potentials.

Question 17:

We have taken the following assumptions in the reduced model:

- h and n roughly have similar behaviour, so h can be approximated as $1-n$
- The decay time of m is very small as compared to h and n . So m has been approximated as $m_{\infty}(V)$ where V is the changing membrane potential.

Using these assumptions, we get the following plots



It can be seen from these images that the cell is firing action potentials just like in four dimensions. But the shape of the action potential is changed. Since we have considered m to be $m_{\infty}(V)$, the rising time for the action potential is almost zero and we have replaced h with $1-n$ so the shape of the drop i.e. the shape of the refractory period is changed and is now dependant on h .

The following code snippet is the differential equation for the reduced system

```
function [ variables ] = deriv_hh(t,states,I_ext)
global G_Na G_K G_l E_Na E_K E_l Q T Cm T_celcius
V=states(1);
n=states(2);
h=1-n;

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

m=alpha_m/(alpha_m+beta_m);

g_Na=G_Na*m^3*h;
I_Na=g_Na*(V-E_Na);

g_K=G_K*n^4;
I_K=g_K*(V-E_K);

g_l=G_l;
I_l=g_l*(V-E_l);

I_ion=I_Na+I_K+I_l;
```



```
alpha_n=-(0.01*(V+50))/(exp(-(V+50)/10)-1);  
beta_n=0.125*exp(-(V+60)/80);  
  
D_n=alpha_n*(1-n)-beta_n*n;  
D_V=(I_ext-I_ion)/Cm;  
  
variables=[D_V;D_n];  
  
return
```