Conductance-based models are the most commonly used models for simulating excitable cells, like neurons. In this paper, we will be looking into the dynamics of a conductance-based model(CBM) called the Morris-Lecar model. Published in 1981, Catherine Morris and Harold Lecar reproduced the oscillatory behaviour of Calcium and Potassium ion conductance in the Barnacle muscle fiber.[1] Although originally based on the Barnacle muscle fiber, the Morris-lecar model can be used to generate action potentials and produce oscillations based on the initial parameters. Additionally we will apply phase-plane analysis and bifurcation theory to examine the dynamics of the system and the behaviour of oscillations as parameters are varied.

1 The Morris-Lecar model

Considered to be a simplified version of the Hodgkin-Huxley model(another type of CBM), the model defines a relationship between the membrane potential V and the activation of calcium and potassium ion channels within the membrane. The model has a few underlying assumptions:

- The calcium ion channels respond to the membrane potential V instantaneously.
- The calcium currents are an inward, non-inactivating current producing a depolarization(excitatory potential) as the potassium currents are an outward, non-inactivating current producing a hyperpolarization(inhibitory potential).
- The calcium system is considered to operate much faster than the potassium system such that g_{Ca} is instantaneously in steady state at all times.

The model is represented in a 2 dimensional system of non-linear differential equations.[1]

$$\begin{aligned} CV' = &-g_{\mathrm{Ca}}M_\infty(V)(V - V_{\mathrm{Ca}}) \\ &- g_{\mathrm{K}}W(V - V_{\mathrm{K}}) - g_{\mathrm{L}}(V - V_{\mathrm{L}}) + I \end{aligned} \tag{1}$$

$$W' = \lambda(V)(W_\infty(V) - W) \tag{2}$$

where V is the membrane potential or voltage(mV), I is the applied current(μA/cm^2), W is the gating variable or the fraction of open potassium ion channels, V_{Ca}, V_K, V_L are the Calcium, Potassium and Leakage voltage constants respectively(mV), g_{Ca}, g_K, g_L are the Calcium, Potassium and Leakage conductance's(mmho/cm^2) respectively and C is the cell capacitance(μF/cm^2) with,

$$M_\infty(V) = (1 + \tanh[(V - V_1)/V_2)])/2 \tag{3}$$

$$W_\infty(V) = (1 + \tanh[(V - V_3)/V_4)])/2 \tag{4}$$

$$\lambda(V) = \phi\cosh[(V - V_3)/2V_4] \tag{5}$$

where V_1, V_2, V_3, V_4(mV) and ϕ(s^{-1}) are tuning parameters.[1] The Calcium and Potassium voltages account for the activation of their respective ion channels. The leak channels account for the membranes natural permeability to ions.

In reference [2] Fig.4, Ashwin et al. use the following set of constants:
$C = 1$, $V_k = -0.7$, $V_L = -0.5$, $V_{Ca} = 1$, $g_K = 2$, $g_L = 0.5$, $g_{Ca} = 1$, $V_1 = -0.01$, $V_2 = 0.15$, $V_3 = 0.1$, $V_4 = 0.145$, $\phi = 1.15$.
The aim of the Morris-Lecar model is to reproduce oscillatory behaviour. We construct a matlab model that uses ode45() to solve the Morris-Lecar model. We then simulate the model for 30 milliseconds(ms) using an applied current of $I = 0$ with initial conditions $[V_0, W_0] = [-0.1, 0]$, and retrieve the following results:

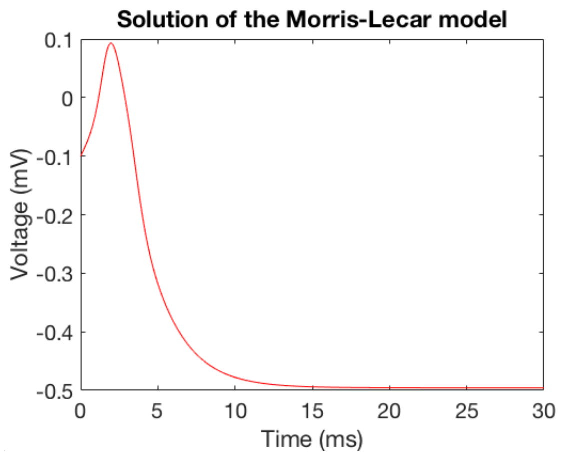

Figure 1 - Voltage solution to the Morris-Lecar model simulated for 30ms.

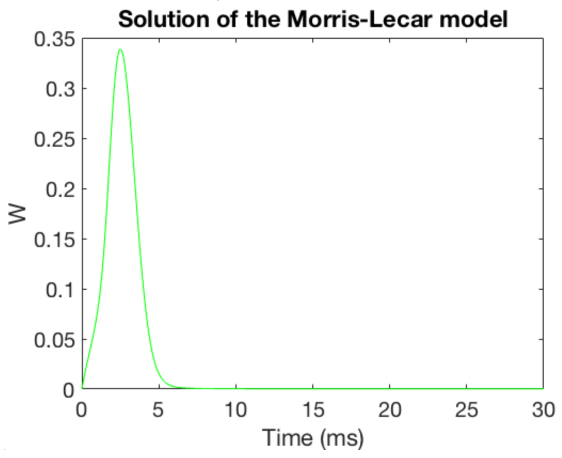

Figure 2 - Gating variable solution to the Morris-Lecar model simulated for 30ms.

For an applied current of $I = 0$, a single peak occurs for both solutions and no oscillations occur. Upon increasing I in increments of 0.005 whilst varying the initial conditions, we find sustained oscillatory behaviour does not occur until $I = 0.075$.

If we simulate the Morris-Lecar model for $I = 0.075$ with initial conditions $[V_0, W_0] = [-0.127, 0.133]$ for 100 ms, we retrieve the following results:

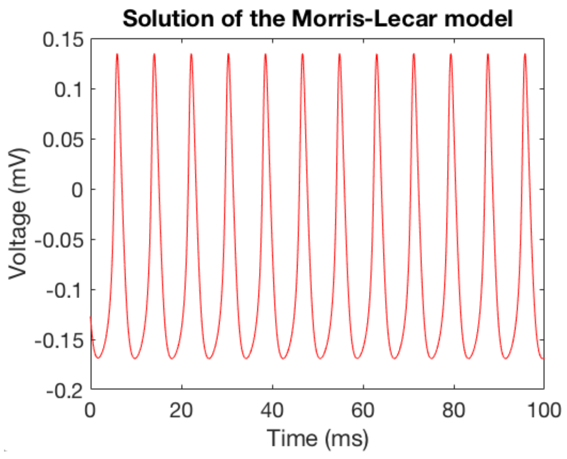

Figure 3 - Voltage solution to the Morris-Lecar model simulated for 100 ms.

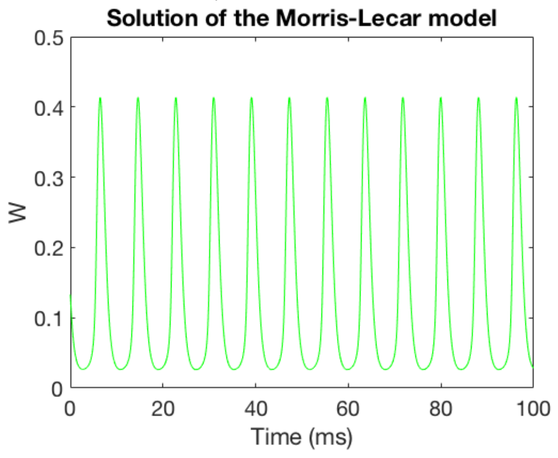

Figure 4 - Gating variable solution to the Morris-Lecar model simulated for 100 ms.

In both figures 3 and 4, the system exhibits sustained oscillatory behaviour after an applied current of 0.075. In reference [2], Ashwin et al. use the same conditions to produce a periodic orbit; so as expected both V and W produce sustained oscillations through time in our solutions. We can examine the behaviour of the orbit by plotting a phase plane portrait and plotting trajectories.

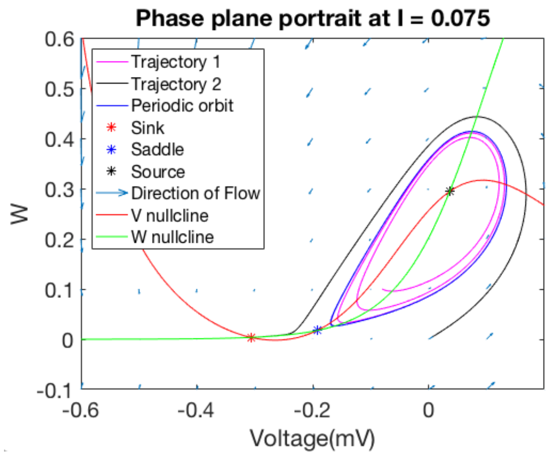

Figure 5 - A phase plane portrait of the Morris-Lecar model; the voltage nullcline is red, the gating variable nullcline is green, the periodic orbit is denoted by the blue line as the alternative trajectories are denoted by the purple and black lines. The fixed points: sink, saddle and source are denoted as red, blue and black asterisks respectively. The arrows denote the direction of flow.

Trajectory 1 tends to the periodic orbit as trajectory 2 follows the orbit around but tends to the sink. Interestingly the periodic orbit(blue) in figure 5 is within very close proximity to the saddle residing at $[V,W]$ = [-0.1919,0.0175]. When decreasing the applied current I slightly, the periodic orbit moves closer to the saddle. Upon collision between the orbit and saddle point; a homoclinic bifurcation occurs.[2] Ashwin et al. cites the homoclinic bifurcation as a way to achieve a lower firing rate.

2 Homoclinic Bifurcation

Bifurcations indicate a sudden change in the qualitative behaviour of a system. A homoclinic bifurcation occurs when a periodic orbit collides with a saddle point and the orbit then becomes a homoclinic orbit.[3] In this case the homoclinic bifurcation I_C acts as a condition for oscillatory behaviour where sustained oscillations are not possible for $I < I_C$. We can deduce from our attempts to produce oscillatory behaviour earlier that I_C may lie in the interval I = [0.07 0.075]. In order to locate the homoclinic bifurcation I_C, we produce a bifurcation diagram using XPPAUT, a numerical tool for simulating and analyzing dynamical systems, which we plot using a custom matlab function(found on XPPAUT website) in figure 6.

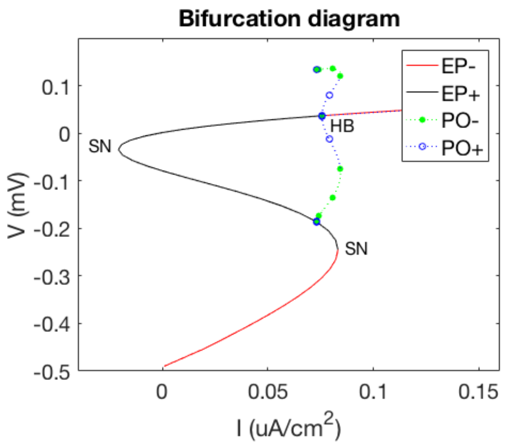

Figure 6 - A bifurcation diagram where the parameter changed is applied current. The red line denotes stable equilibria and the black line denotes unstable equilibria. The filled-green circles denote the extrema of stable periodic orbits as the blue circles denote the extrema of unstable periodic orbits. HB refers to the Hopf bifurcation and SN refers to the saddle-node bifurcations.

Unlike local bifurcations like saddle-node bifurcations(SN) or hopf bifurcations(HB), a homoclinic bifurcation cannot be found purely through stability analysis of equilibria as it is a global bifurcation. Instead we must additionally compute the periodic orbits off the Hopf bifurcation via numerical continuation. In figure 6 we observe both periodic solution branches terminate at the same value of I which is the location of the homoclinic bifurcation. Curiously we see some interesting dynamics where the period scales massively as the bifurcation is approached. In reference [2], Ashwin et al. state that the "frequency of oscillation scales as $-\frac{1}{\log(I-I_C)}$." This means that as I approaches I_C from infinity the frequency decreases sharply to zero. Since $Frequency = 1/Period$ then the period must tend to infinity as I approaches I_C which explains the dynamics we see in our periodic solution branches. Strogatz [3] also refers to the homoclinic bifurcation as an infinite-period bifurcation for that reason. We can take the periodic solution data from XPPAUT and plot the period of the orbits against I to find an approximation of I_C where the period is at its largest.

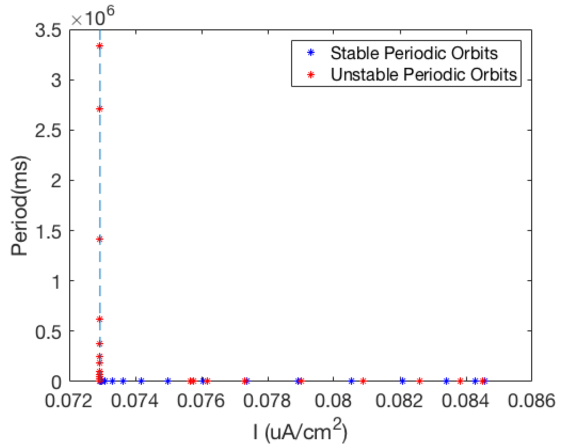

Figure 7 - A plot of the period of periodic orbits against the applied current I. The dashed line is our approximation of $I = I_C$.

Locating the orbit point of highest period that XPPAUT was able to compute as shown by the dashed line; our respective approximation is $I_C \approx 0.072932$. Choosing an applied current I close to the true value I_C but larger, results in the model producing a larger period and so a lower firing rate as desired. In Figure 8 we observe the voltage solution at $I = 0.072932$ which has a lower firing rate(frequency of oscillations) than in figure 3 when $I = 0.075$. Thus we can conclude that using an applied current that is $I > I_C$ and closer to the homoclinic bifurcation will produce a lower firing rate.

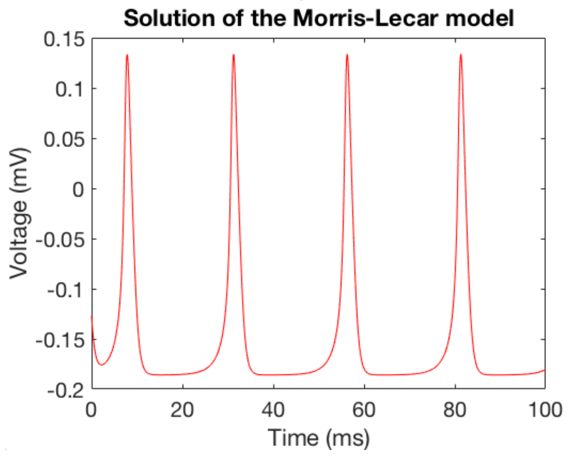

Figure 8 - Voltage solution to the Morris-Lecar model simulated for 100ms using the same constants as Ashwin for an applied current of I = 0.072932

3 Additional Remarks

There are 2 types of homoclinic bifurcation; for which the type depends on the stability of the equilibrium centered within the homoclinic orbit. As can be seen in our bifurcation diagram we know that this equilibrium is a source(unstable) for values of I before and after the homoclinic bifurcation. If we take a look at the phase plane graph for an applied current $I = 0.07$, we can observe the dynamics to identify the homoclinics type.

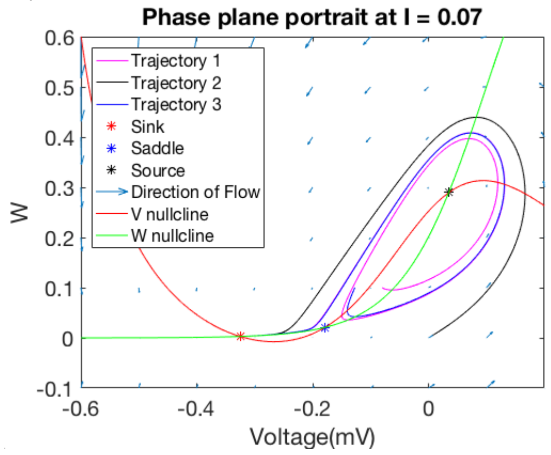

Figure 9 - A phase plane portait of the Morris-Lecar model at $I = 0.07$ with multiple trajectories.

We observe that for $I < I_C$ the limit cycle is broken as expected and all trajectories tend to the sink. This leads me to classify I_C as a supercritical homoclinic bifurcation.[4]

References

[1] Catherine Morris and Harold Lecar. Voltage oscillations in the barnacle giant muscle fiber. *Biophysical journal*, 35(1):193–213, 1981.

[2] Peter Ashwin, Stephen Coombes, and Rachel Nicks. Mathematical frameworks for oscillatory network dynamics in neuroscience. *The Journal of Mathematical Neuroscience*, 6, 2016.

[3] Steven H Strogatz. Non-linear dynamics and chaos. *Studies in Non-linearity*, page 262, 1994.

[4] Eugene M Izhikevich. *The Dynamical Systems in Neuroscience: Geometry of Excitability and Bursting*. MIT Press, 2007.

Appendix

Matlab and XPPAUT code can be found at the back. Figures 1,2,3,4,5 and 8 were produced using ”morris_lecar.m” ,”mlsolve.m”, ”MyJacobian.m”, ”MySolve.m”. Figures 6 and 7 can be produced using a culmination of XPPAUT and Matlab files: ”MorrisLecar.ode”, ”periodlplot.m”,”pdata.dat”(not attached). All files can be found at `github.com/sahilkariadata/morris_lecar` including ”pdata.dat”.

'morris_lecar.m'

```
%% initialised variables
ml.I = 0.075;
ml.C = 1;
ml.VK = -0.7;
ml.VL = -0.5;
ml.VCa = 1;
ml.gK = 2;
ml.gL = 0.5;
ml.gCa = 1;
ml.V1 = -0.01;
ml.V2 = 0.15;
ml.V3 = 0.1;
ml.V4 = 0.145;
ml.phi = 1.15;

%% initial conditions
v0 = -0.127;
w0 = 0.133;
y0 = [v0;w0];

%% solve the DE's using ode45
t0 = 0;
tend = 100;
tspan = [t0 tend];
options = odeset('Abstol',1e-8,'RelTol',1e-6);
[t,y] = ode45(@mlsolve,tspan,y0,options,ml);
[t2,y2] = ode45(@mlsolve,tspan,[-0.08,0.1],options,ml); %alt. trajectory 1
[t3,y3] = ode45(@mlsolve,tspan,[0,0],options,ml); %alt. trajectory 2

%% plot both v and w against time
figure(1);
plot(t,y(:,1),'r');
title('Solution of the Morris-Lecar model');
xlabel('Time (ms)');
ylabel('Voltage (mV)');
set(gcf,'color','w');
set(gca,'fontsize',20);
figure(2);
plot(t,y(:,2),'g');
title('Solution of the Morris-Lecar model');
xlabel('Time (ms)');
ylabel('W');
set(gcf,'color','w');
set(gca,'fontsize',20);

%% set up equations for phase plane portrait
minv = @(v) 0.5*(1+tanh((v-ml.V1)/ml.V2));
winv = @(v) 0.5*(1+tanh((v-ml.V3)/ml.V4));
lambda = @(v) ml.phi*cosh((v-ml.V3)/(2*ml.V4));
dvdt = @(v,w) (1/ml.C)*(ml.gL*(ml.VL-v) + ml.gK*w.*(ml.VK-v) + ...
    ml.gCa*minv(v).*(ml.VCa-v) + ml.I);
dwdt = @(v,w) lambda(v).*(winv(v) - w);

[X,Y] = meshgrid(-0.6:0.1:0.6); %set up grid for quiver
DV = dvdt(X,Y);
DW = dwdt(X,Y);
fun = @(y) mlsolve(t,y,ml); %set up function to find fixed points
h = 1e-6;
df = @(y) MyJacobian(fun,y,h);

saddle = MySolve(fun,[-0.2;0],df,1e-6,100); %custom function that uses MyJacobian ...
    to solve for equilibria
sink = MySolve(fun,[-0.4;0],df,1e-6,100);
source = MySolve(fun,[0;0],df,1e-6,100);
```

```

figure(3);
plot(y2(:,1),y2(:,2),'m',y3(:,1),y3(:,2),'k'); %plots trajectories 1 and 2
hold on
plot(y(:,1),y(:,2),'b'); %plots main trajectory(blue)
title('Phase plane portrait at I = 0.075');
xlim([-0.6 0.2]);
ylim([-0.1 0.6]);
xlabel('Voltage(mV)')
ylabel('W')
hold on
plot(sink(1),sink(2),'r*',saddle(1),saddle(2),'b*',source(1),source(2),'k*','MarkerSize',8) ...
    %plots fixed points
hold on
quiver(X,Y,DV,DW) %Plots phase plane arrows(direction of flow)
hold on
fcontour(dvdt,[-1 1 -1 1],'-r','LevelList',[0],'MeshDensity',200) %V Nullcline found
hold on
fcontour(dwdt,[-1 1 -1 1],'-g','LevelList',[0],'MeshDensity',200) %W Nullcline found
set(gcf,'color','w');
set(gca,'fontsize',20);
legend({'Trajectory 1','Trajectory 2','Periodic ...
    orbit','Sink','Saddle','Source','Direction of Flow','V nullcline','W ...
    nullcline'},'Location','northwest','Fontsize',16)
hold off
```

'mlsolve.m'

```
function dydt = mlsolve(t,y,ml)

minv = @(v) 0.5*(1+tanh((v-ml.V1)/ml.V2)); %computes ion channel functions
winv = @(v) 0.5*(1+tanh((v-ml.V3)/ml.V4));
lambda = @(v) ml.phi*cosh((v-ml.V3)/(2*ml.V4));

dvdt = (1/ml.C)*(ml.gL*(ml.VL-y(1)) + ml.gK*y(2)*(ml.VK-y(1)) + ...
    ml.gCa*minv(y(1))*(ml.VCa - y(1)) + ml.I);
dwdt = lambda(y(1))*(winv(y(1)) - y(2));
dydt = [dvdt;dwdt]; %outputs gradient of dV/dt and dW/dt
end
```

'MyJacobian.m'

```
function df = MyJacobian(f,x,h)
len = length(x);
xs = x;
xa = x;
%% carries out centred difference to evaluate f'(x)
for i = 1:len
    xs(i) = xs(i) + h;
    xa(i) = xa(i) - h;
    df(:,i) = (f(xs) - f(xa))/(2*h);
    xs(i) = x(i);
    xa(i) = x(i);
end
end
```

'MySolve.m'

```
function [x,conv,J] = MySolve(f,x0,df,tol,maxit)
if maxit ≠ 0 %if maxit is zero the convergence hasn't worked
    J = df(x0);
    x1 = x0 - J\f(x0);
    if norm(abs(x1-x0)) && norm(f(x0)) < tol
        conv = 1;
        x = x1;
        return
    else
        maxit = maxit - 1; %uses recursion by decreasing maxit
        [x,conv,J] = MySolve(f,x1,df,tol,maxit); %uses newest approximation
    end
else
    conv = 0;
    x = 0;
    J = 0;
    return
end
end
```

'periodlplot.m'

```
data = importdata('pdata.dat'); %imports periodic orbit data from xppaut .dat file
data(data(:,5)≠4,:) = []; %gets rid of all data not periodic orbits
I = data(:,1); %applied current for each orbit
spo = data;
upo = data;
spo(spo(:,4)≠3,:) = nan; %singles out the stable orbits
upo(upo(:,4)≠4,:) = nan; %singles out the unstable orbits
spo = spo(:,2);
upo = upo(:,2); %period values

plot(I,spo,'b*',I,upo,'r*') %plots the stable and unstable orbits
hold on
line([0.072932 0.072932], [0 3500000],'LineStyle','--'); %plots a dashed line at I ...
    = Ic
%closer approximation I_c = 0.07293189486804816 as xppaut rounds upon conversion ...
    to .dat file
xlabel('I (uA/cm^2)')
ylabel('Period(ms)')
%ylim([0 3.5])
% title('Plot of period of orbits against applied current I')
legend('Stable Periodic Orbits','Unstable Periodic Orbits')
set(gcf,'color','w');
set(gca,'fontsize',20);
```

'MorrisLecar.ode'

```
p I=0.075

v(0)=-0.127
w(0)=0.133

gca=1.0
gk=2.0
gl=0.5
vk=-0.7
```

```
vl=-0.5
f=1.15

v1=-0.01
v2=0.15
v3=0.1
v4=0.145

minf=0.5*(1.0+tanh((v-v1)/v2))
winf=0.5*(1.0+tanh((v-v3)/v4))
tauw=1.0/cosh((v-v3)/(2.0*v4))

v'=-gca*minf*(v-1.0)-gk*w*(v-vk)-gl*(v-vl)+I
w'=f*(winf-w)/tauw

@ xp=v,yp=w,xlo=-0.5,xhi=0.2,ylo=-0.1,yhi=0.5
@ bounds=100000,maxstore=100000,nmesh=200,total=10.0
@ method=runge-kutta,dt=0.001
```