ES 226R Problem Set 1

Andrew T. Sullivan

3 March 2020

1 Question 1: The Hodgkin-Huxley Model

In order to simulate the Hodgkin-Huxley model of action potential propagation, two functions were constructed in Matlab. The first takes in a time variable, an input vector containing the state of the system in terms of membrane potential and the values of the three gating variables, and the injected current. This function uses these variables, along with specified conductances, Nernst potentials, and forms of the a and b equations for each gating variable to evaluate the differential equations governing how each state variable changes in time. The output of the function contains the time derivatives of all four state variables. The second function is broken up into seven parts. In the first, the potential is set equal to the resting membrane potential of the cell of -61.2 mV, the time vector is produced over which the dynamics are evaluated, and the initial conditions for the gating variables are introduced, based on solving the asymptotic voltage/infinity curves for each variable at the resting membrane potential. The conductances and Nernst potentials are also defined here for use in subsequent parts.

1.1 a)

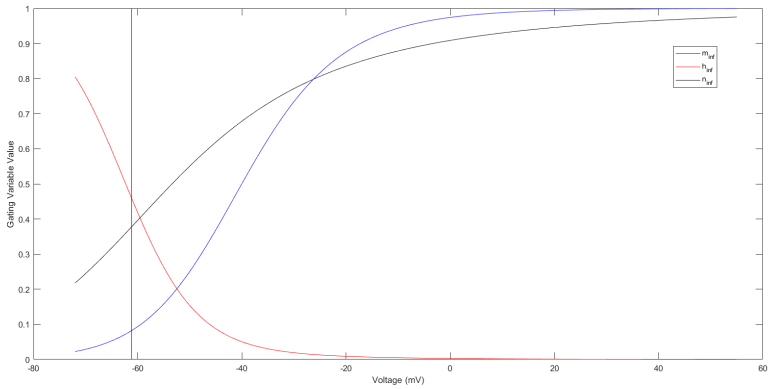


Figure 1: Voltage Dependence of m_∞, h_∞, and n_∞ Curves.

The first figure for the Hodgkin-Huxley model contains the voltage dependence of the asymptotic curves for each of the three gating variables, evaluated using the functional forms specified above. The voltage range is chosen to be between the Nernst potential for potassium, -72 mV, and for sodium, +55 mV. As expected, the activation variables, m and n, increase as a function of cell depolarization, while the inactivation variable, h, decreases. The resting membrane potential is indicated by the black vertical line. At this state, most sodium channels are closed or inactivated, reflecting the substantially lower membrane conductance for sodium at rest compared to potassium.

1.2 b)

The next figure shows the influence of injected current on the model. We evaluate this response by initializing the current as an anonymous function of time with fixed magnitude for each iteration, and initialize the state variables as defined above. For each current, a fourth-order Runge-Kutta method is used to evaluate the system of four coupled differential equations at all time points, and the results are plotted. Each subplot shows the membrane response to a different injected current density, ranging from 1 to 10 $\mu A/cm^2$. In the top-left subplot, it is apparent that the membrane voltage fluctuates briefly, but this low current density is insufficient to produce a suprathreshold depolarization and will not initiate an action potential. Subsequent larger current densities all produce action potentials, with the time of the first spike progressively decreasing as threshold is reached faster, and the firing frequency increasing for subsequent spikes as well. Though this is the expected qualitative behavior, constant current input is typically not a biologically relevant condition.

1.3 c)

The third figure for this model explores the response to a pulse of current, rather than a sustained injection, bringing the situation closer to the reality in *in vivo* neural systems. The current is initialized such that it has a value of $10\mu A/cm^2$ between 5 and 6 msec, and 0 elsewhere.The shape of the action potential is similar to those in the previous

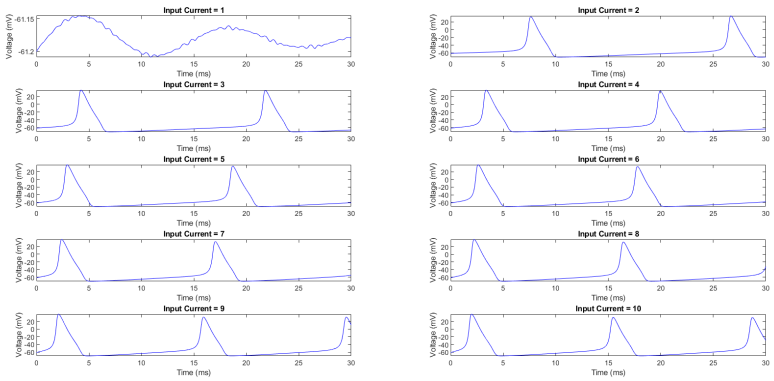


Figure 2: Evolution of Membrane Potential in Response to Constant Current Densities.

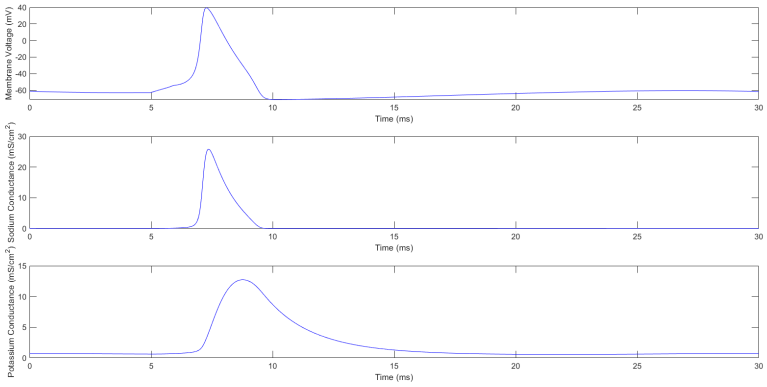


Figure 3: Evolution of Membrane Potential and Conductances in Response to 1 msec Current Pulse of Magnitude 10 $\mu A/cm^2$.

figure, with a slow initial increase as threshold is approached, followed by a transient upstroke, prolonged downstroke, and afterhyperpolarization. Examining the following two subfigures reveals the contribution of conductances to these behaviors. Here, the conductance is plotted as $G_{Na}m^3h$ for sodium and $G_K n^4$ for potassium. We can see that, like the action potential, the sodium conductance rapidly increases to a maximum. Around this time, the action potential also reaches its maximal value, approaching but not reaching the sodium equilibrium potential. The sodium conductance then begins to decrease as the slower inactivation variable decreases, and the potassium conductance increases, counteracting the sodium influx with potassium efflux. The sodium conductance returns approximately to its baseline value after a few msec, but the potassium remains elevated for several milliseconds later, producing the sustained afterhyperpolarization and refractory period characteristic of action potentials.

1.4 d)

We next explore the activation function of the Hodgkin-Huxley neuron, characterized by the relationship between peak membrane current and the input current magnitude of a 1 msec pulse. We initialize the current using the same scheme as the previous part, but changing the magnitude of the pulse for each iteration of the loop. The maximal value is extracted from each voltage vs. time trace and stored to produce the activation function. This function is sigmoidal in nature and is approximately a logistic function, consistent with other models of neural activation. Below input current pulses of approximately $4\mu A/cm^2$, small subthreshold depolarizations are all that result from input. Above this value, the neuron fires an action potential with nearly constant amplitude. While the action potential is typically regarded as an all-or-none phenomenon, this function reveals a small deviation from step-function behavior at current densities very close to the threshold. It should be noted that these are currents of fixed density and duration, and thus changes in area or time will produce different thresholds, all of which correspond to approximately the same voltage threshold.

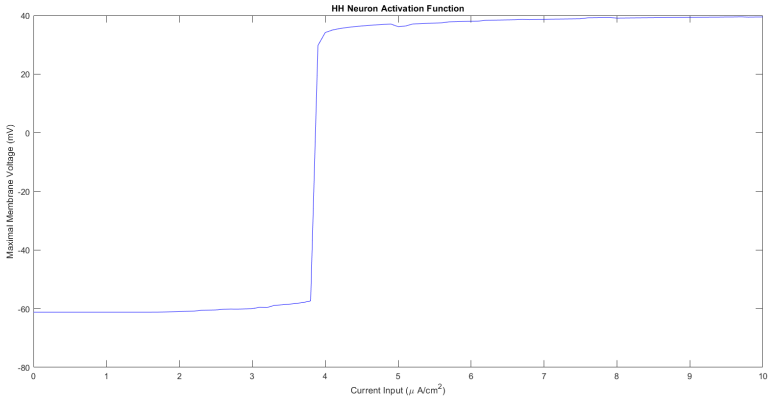


Figure 4: Relationship Between Current Magnitude and Peak Membrane Voltage for 1 msec Pulse.

1.5 e)

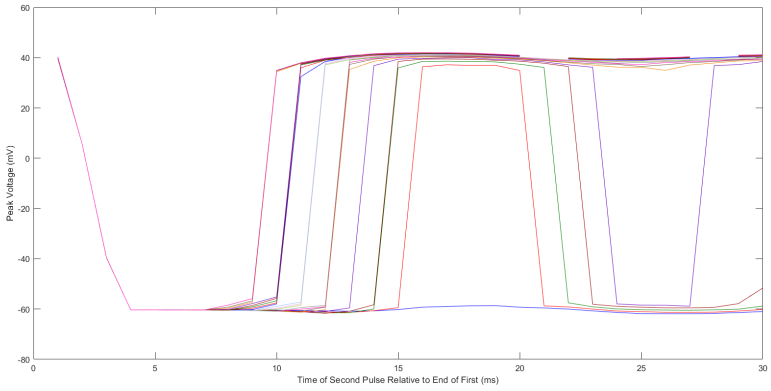
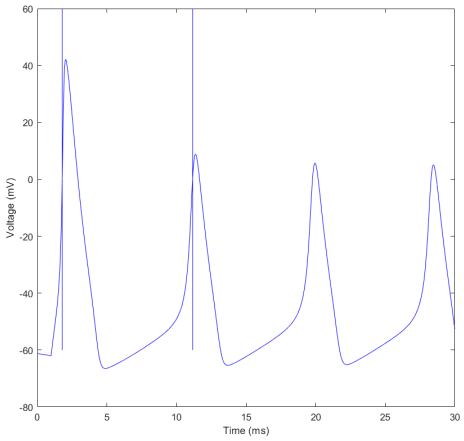


Figure 5: Relationship Between Time After First 1 msec-10 $\mu A/cm^2$ Pulse and Peak Membrane Voltage for 1 msec Pulses of Varying Magnitude.

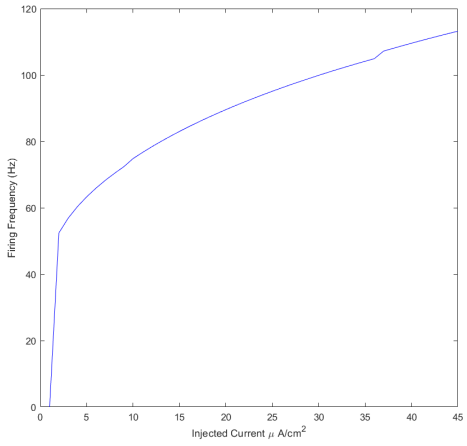
In order to explore the influence of the refractory period, we begin all simulations with a 1 msec-10 $\mu A/cm^2$ pulse, as seen before. We add to this a second pulse of varying magnitude at a varying time after the first pulse. For a given magnitude, we simulate the resultant voltage to a total of 40 msec with the second pulse coming between 1 and 30 msec after the end of the first pulse. We perform this simulation for current magnitudes between 1 and 20 $\mu A/cm^2$. From each trace, we extract the maximum voltage attained after the end of the first pulse, and plot each peak voltage-time after the first pulse trace for all magnitudes. In this plot, the smallest magnitude is the curve which does not ever pass above -50 mV, except within the first 3 msec, since these maximal voltages instead reflect the first action potential. As currents get larger, the curves increase in magnitude along with the portion of the curve reflecting suprathreshold depolarization. The next largest curve only shows the initiation of a second action potential with pulses 15 msec after the first, while the largest magnitude can initiate a second spike only 9 msec after the first pulse ends, reflecting the so-called "relative refractory period." Interestingly, lower-amplitude pulses show a return to subthreshold depolarization around 20-25 msec after the first pulse. Investigation of the voltage traces shows the presence of small potential oscillations around this value, which are sufficient to counteract the influence of the input current if this current is small enough in magnitude.

1.6 f)

The final figure in the Hodgkin-Huxley model captures the voltage dependence of neuronal firing rate. We simulate this by first declaring a threshold above which we qualify a membrane depolarization as a spike (here, set to 0 mV). The relevant current density range is chosen to be 1 to 45 $\mu A/cm^2$, since values above this range no longer



(a) Method for Calculating Firing Frequency.



(b) Firing Frequency.

Figure 6: Relationship Between Firing Rate and Current Magnitude for Constant Current Densities.

produce meaningful results (no spikes are seen and the model appears to diverge, which may be a consequence of the solver being used). For each current density, current is injected beginning at 1 msec through the end of the simulation (30 msec). We extract spikes from the voltage trace by finding all values of the voltage that are above the threshold. From these indices (if they exist), we find all indices where the difference to the next index is not equal to 1, i.e. corresponding to the end of one spike and the beginning of the next. For each of these indices, we take the one beginning the following spike. We also add on the first value in the array, corresponding to the onset of the first spike. For each spike for a given density, we extract the interspike intervals and average their values. This is schematically depicted in the first subfigure above for the first two spikes. If there are zero or one detected spike in the range, we set the frequency equal to zero. For all other cases, the firing frequency is taken as the reciprocal of the average interspike interval. This is plotted as a function of input current density, revealing the presence of an initial threshold, as seen before, and a sublinear relationship between firing rate and input current above the initial linear regime. Most neurons do not fire at such high frequencies (above 100 Hz), but instead stay closer to the most sensitive dynamical regime (around 60 Hz or less).

2 Question 2: Phase Plane Analysis

Given that simulation of a system of four coupled differential equations is rather challenging, and interpreting the results in this higher-dimensional space can be equally as difficult, simplifications are commonly used to lower the dimensionality of the Hodgkin-Huxley system to permit standard dynamical systems techniques to be applied. Two such simplifications are presented and analyzed below. The first makes use of the observation that the time course of the m-gate sodium conductance is much faster than the n- or h-gates. Therefore, we can assume that the m-gate reaches its asymptotic value, m_∞, at each value of the membrane voltage, removing one differential equation. We also make the assumption that the h- and n-gates are linearly related, here by the equation $h + n = 0.8$, so that solving one of the differential equations yields the solution to the other. With this methodology, we can reduce the four dimensions of the system (plus time) down to two and perform phase-plane analysis, examining the dynamics of the system as they evolve over time as well as on a two-dimensional plane containing both the time-dependent variables. The function for this updated model takes in a time vector, a current vector, and a two-dimensional input vector containing the current state of the voltage and n variables. From n, h is found from the linear equation, and the voltage is used to evaluate the asymptotic value of m, which is set equal to the state variable due to the above assumption. The voltage is also used to evaluate the α_n and β_n equations, which are used to find the derivative of n as $\alpha_n(1-n) - \beta_n n$. Finally, the voltage derivative is evaluated.

2.1 a)

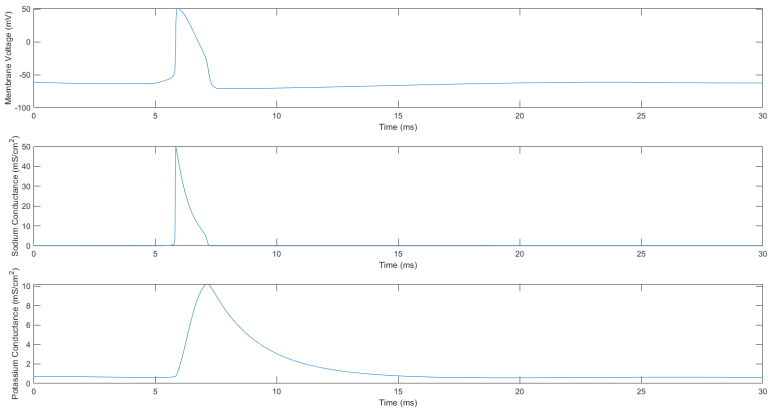


Figure 7: Evolution of Simplified Membrane Potential and Conductances in Response to 1 msec Current Pulse of Magnitude 10 $\mu A/cm^2$.

We first examine the temporal dependence of the system in response to the same current pulse described in Figure 3. Comparing Figure 7 to Figure 3, we can see that the updated model does a fairly good job of reproducing the behavior of the action potential, though the maximum attained voltage is slightly larger and the action potential duration is reduced. The sodium conductance is noticeably different, as the instantaneous opening produces a sharp spike in conductance with a sharp drop-off, rather than the rounded peak found in the full model. Finally, the potassium conductance evolution is somewhat sharper in the reduced model, with a smaller final value. Overall, the temporal behavior is mostly similar.

2.2 b)

We can also compare differences in the phase plane behavior of the two models, on a potassium conductance-vs.-membrane voltage plane. Again, we see qualitative similarity between the two models, with overall differences in the magnitudes. The simplified model reaches a larger maximum membrane voltage, likely due to the rapid sodium response, and the maximum achieved potassium conductance is smaller, as found in the temporal plots. The initial onset is more rapid for the simplified model, as evidenced by the small change in potassium conductance as voltage travels up to the maximum, though the final repolarization phase is comparable. Again, we find that this model does a good job of recapitulating the important behaviors found in the full model.

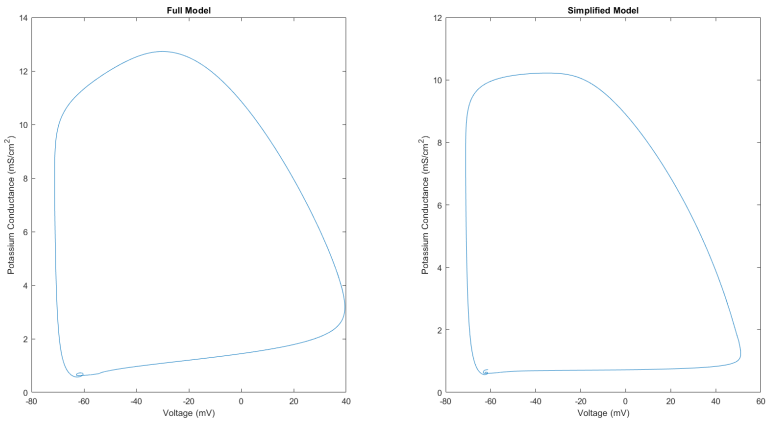
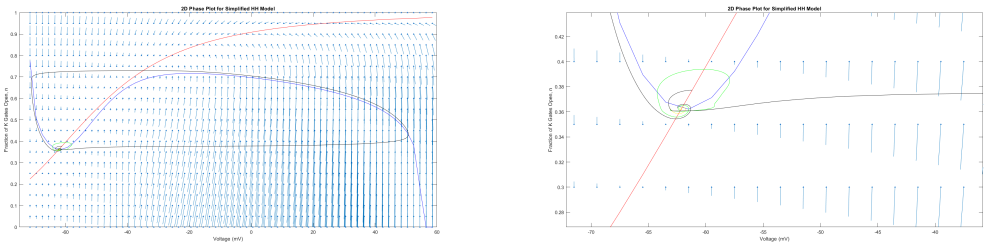


Figure 8: Comparison of Phase Plane Behavior Between the Full (left) and Simplified (right) Models.

2.3 c)



(a) 2D Phase Plot Vector Field.

(b) Zoom-in of a Around the Resting Potential.

Figure 9: Phase-Plane Behavior of Simplified Model.

A final means of characterizing the behavior of the system is in terms of a vector plot using the "quiver" function in Matlab, which plots the derivatives (outputs of the model) as vectors on the phase plane, thus providing insight into how the system would evolve at any 2D point in its state space. This plot also contains the nullclines of the system, i.e. the manifolds on which one of the derivatives in the system is zero. These are locations where the derivative can change sign. The intersections of these manifolds are called fixed points and are locations at which the system would remain if all other outside influences were removed. These fixed points can be attractive or repulsive depending on the nearby derivatives. The n nullcline (red) is found quite simply by setting $n = n_\infty$. The V nullcline (blue) is a bit more challenging. Ultimately, we seek to solve the equation $G_{Na}m_\infty^3(0.8-n)(V-E_{Na})+G_K n^4(V-E_K)+G_L(V-E_L) = 0$, which, given m_∞'s dependence on V, is not readily solved for the membrane voltage. However, we can note that this equation is a fourth-order polynomial in n, and as such can be solved and evaluated for n at each voltage value. This is accomplished using Matlab's "roots" function, and extracting the positive real root (if it exists). Finally, sample trajectories are plotted in response to a 1 msec pulse of amplitude 0, 4, or 8$\mu A/cm^2$ (yellow, green, and black curves). The presence of a yellow curve suggests that the starting values chosen for V and n are not the steady-state values for this system, since it will evolve even without external input. The green curve shows a small oscillation around this equilibrium point, evidence that this is a subthreshold depolarization. Finally, the black curve shows a suprathreshold depolarization, which rapidly approaches the V nullcline and evolves along this trajectory until the opening of potassium channels pulls the state back towards the resting potential.

2.4 d)

We now consider a different simplification to the Hodgkin-Huxley model. Above, it was noted that the sodium conductance, and in response, the membrane voltage, approached their maximum values unrealistically rapidly due to our assumption that the m variable instantaneously approaches its asymptotic value. A separate assumption that can be made to address this shortcoming is that the two "slow" variables, n and h, are so slow that they are

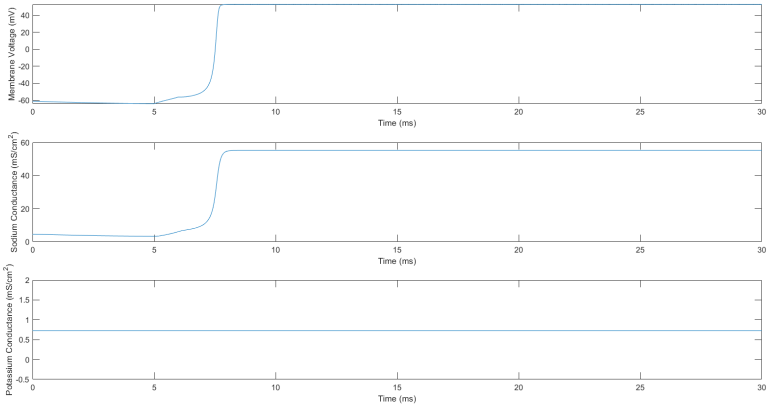


Figure 10: Evolution of Simplified Membrane Potential and Conductances in Response to 1 msec Current Pulse of Magnitude 10 $\mu A/cm^2$.

essentially constant with respect to m and V. Thus, if we set n and h equal to their initial values throughout the simulation, we can examine action potential onset in more detail.

Our temporal simulation shows the expected long-term behavior - without the closing of sodium channels due to h inactivation and opening of potassium channels due to n, the membrane potential approaches a constant steady-state value. However, we can also see that the upswing of the membrane potential in this model, as well as the initial evolution of the sodium conductance, more accurately capture the dynamics exhibited in the full model in comparison to our previous case, beginning with a slow increase to threshold potential followed by a rapid (but not instantaneous) upswing.

2.5 e)

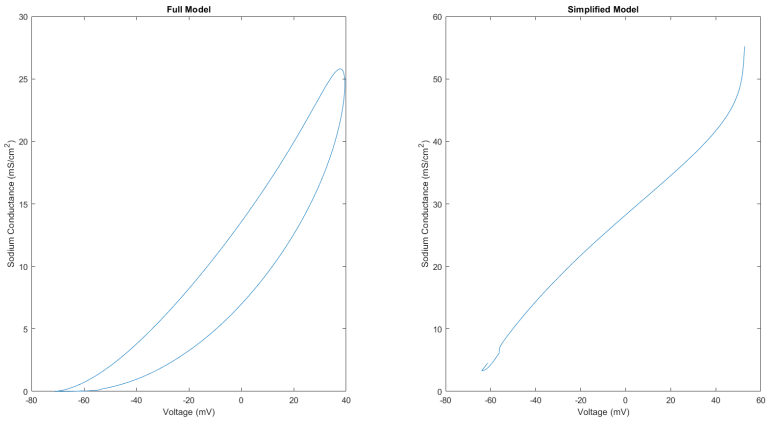
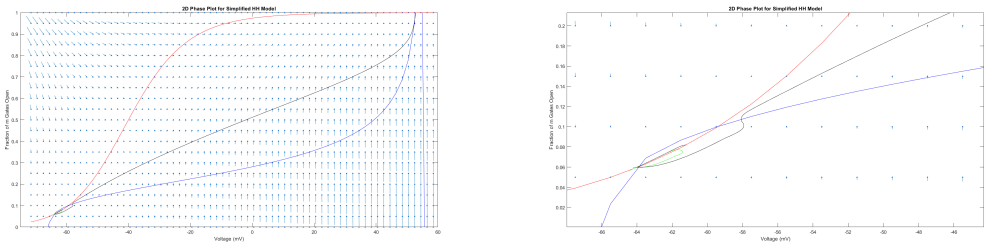


Figure 11: Comparison of Phase Plane Behavior Between the Full (left) and Simplified (right) Models.

Rather than consider the $n - V$ phase plane, since n does not evolve in this model, we consider the $m - V$ phase plane. Similarly to above, we see a rapid exponential upswing in the simplified model to the new steady-state voltage value. Interestingly, an inflection exists in this trajectory that is not found in the full model, which exhibits more obvious exponential behavior. It's possible that this is due to the non-equilibrium nature of the starting points for the simplified model, as the steady-state is likely different than that found in the full case, and these steady-state values are used as the initial conditions. This may explain the initial oscillations and deviations from exponential increase.

2.6 c)



(a) 2D Phase Plot Vector Field. (b) Zoom-in of a Around the Resting Potential.

Figure 12: Phase-Plane Behavior of Simplified Model.

Finally, we examine the phase-plane behavior including evolution vectors, nullclines, and sample trajectories. As before, the m nullcline is simply the line where $m = m_\infty$, determined by evaluating the α_m and β_m equations at each voltage point. The V nullcine is found as the roots of the third-order polynomial equation in m with coefficients $G_{Na}h_0(V - E_{Na})$, 0, 0, and $G_K n_0^4(V - E_K) + G_L(V - E_L)$ at each voltage value. This model exhibits 3 fixed points, but the intermediate one is a repellor, while the lowest (resting membrane potential) and highest (generation of an action potential) are attractors. Figure 12b shows a zoomed-in region around the lower two fixed points showing the cell returning to the resting potential if no current is injected or for a subtreshold depolarization, while the black curve shows the generation of an action potential. This bistability is not exhibited in either of the other two models since they both exhibit mechanisms for repolarization. Clearly, the overall dynamics of the action potential are more accurately captured by the slow/fast model, but this fast-only model provides insight into the behavior of action potential generation absent from the previous version. Thus, multiple phase plane projections could be used to examine different aspects of electrogenic behaviors in neurons.

3 Appendix

The following code is the Hodgkin-Huxley function evaluator, which evaluates the four coupled differential equations based on the state of the system:

```
function [Output] = HH(t,in,I)
E = in(1);
m = in(2);
h = in(3);
n = in(4);
Output = zeros(4,1);

G_Na = 120;
G_K = 36;
G_L = 0.3;
E_Na = 55;
E_K = -72;
E_L = -50;

Output(1) = -G_Na.*m^3.*h.*(E-E_Na)-G_K.*n^4.*(E-E_K)-G_L.*(E-E_L)+I(t);

am = @(x) -0.1*(40+x)/(exp(-(40+x)/10)-1);
bm = @(x) 4*exp(-(65+x)/18);

ah = @(x) 0.07*exp(-(x+65)/20);
bh = @(x) 1/(exp(-(35+x)/10)+1);

an = @(x) -0.01*(55+x)/(exp(-(55+x)/10)-1);
bn = @(x) 0.125*exp(-(x+65)/80);

Output(2) = am(E)*(1-m)-bm(E)*m;
Output(3) = ah(E)*(1-h)-bh(E)*h;
Output(4) = an(E)*(1-n)-bn(E)*n;

tm = 1/(am(E)+bm(E));
minf=am(E)/(am(E)+bm(E));
th = 1/(ah(E)+bh(E));
hinf = ah(E)/(ah(E)+bh(E));
tn = 1/(an(E)+bn(E));
ninf = an(E)/(an(E)+bn(E));
```

The following script is used to solve all six parts of the Hodgkin-Huxley model question:

```
E = -61.2;
figure;
m0=0.0820;
h0=0.4603;
n0=0.3772;
tinc=0.001;
tspan=0:tinc:30;

G_Na = 120;
G_K = 36;
G_L = 0.3;
E_Na = 55;
E_K = -72;
E_L = -50;

%% Part 1
```

```
Vrange = [-72:0.01:55];
for i = 1:length(Vrange)
    am = -0.1*(40+Vrange(i))/(exp(-(40+Vrange(i))/10)-1);
    bm = 4*exp(-(65+Vrange(i))/18);

    ah = 0.07*exp(-(Vrange(i)+65)/20);
    bh = 1/(exp(-(35+Vrange(i))/10)+1);

    an = -0.01*(55+Vrange(i))/(exp(-(55+Vrange(i))/10)-1);
    bn = 0.125*exp(-(Vrange(i)+65)/80);

    tm = 1/(am+bm);
    minf(i)=am/(am+bm);
    th = 1/(ah+bh);
    hinf(i) = ah/(ah+bh);
    tn = 1/(an+bn);
    ninf(i) = an/(an+bn);
end

plot(Vrange,minf);
hold on;
plot(Vrange,hinf);
plot(Vrange,ninf,'k');
yl = ylim;
line([E E],yl,'Color','k');
xlabel('Voltage (mV)');
ylabel('Gating Variable Value');
legend('m_{inf}','h_{inf}','n_{inf}');
hold off;

%% Part 2

figure;
for k = 1:10
    I = @(t) k;
    y0 = [E m0 h0 n0];
    [T,Y] = ode45(@(t,in) HH(t,in,I), tspan, y0);
    subplot(5,2,k);
    plot(T,Y(:,1));
    xlabel('Time (ms)');
    ylabel('Voltage (mV)');
    plottitle = ['Input Current = ',num2str(k)];
    title(plottitle);
end

%% Part 3

figure;
I = @(t) (t>=5 && t<6)*10;
[T,Y] = ode45(@(t,in) HH(t,in,I), tspan, y0);

subplot(3,1,1);
plot(T,Y(:,1));
xlabel('Time (ms)');
ylabel('Membrane Voltage (mV)');

for i = 1:length(T)
```

```
    g_Na(i) = G_Na*Y(i,2)^3*(Y(i,3));
    g_K(i) = G_K*Y(i,4)^4;
    %g_Na(i) = Y(i,3);
    %g_K(i) = Y(i,4);
    IT(i)=I(T(i));
end

% subplot(3,1,1);
% plot(T,IT);
% xlabel('Time (ms)');
% ylabel('Input Current (\mu A/cm^2)');

subplot(3,1,2);
plot(T,g_Na);
xlabel('Time (ms)');
ylabel('Sodium Conductance (mS/cm^2)');

subplot(3,1,3);
plot(T,g_K);
xlabel('Time (ms)');
ylabel('Potassium Conductance (mS/cm^2)');

%% Part 4
figure;

Irange = 0:0.1:10;
for k = 1:length(Irange)
    I = @(t) (t>=5 && t<6)*Irange(k);
    [T,Y] = ode45(@(t,in) HH(t,in,I), tspan, y0);
    Vmax(k) = max(Y(:,1));
end

plot(Irange,Vmax);
xlabel('Current Input (\mu A/cm^2)');
ylabel('Maximal Membrane Voltage (mV)');
title('HH Neuron Activation Function');

%% Part 5
figure;

Irange2 = 1:20;
trange = 1:30;
step = 0.001;
tspan2 = 0:step:40;
for k = 1:length(Irange2)
    for l = 1:length(trange)
        I = @(t) (t>=5 && t<6)*10 + (t>=6+trange(l) && t<6+trange(l)+1)*Irange2(
            k);
        [T,Y] = ode45(@(t,in) HH(t,in,I), tspan2, y0);
        Vmax2(k,l) = max(Y(((6+trange(l))/step):end,1));
%         if k == 2
%             plot(T,Y(:,1));
%             hold on;
%             disp(l);
%             pause;
%
%         end
```

```
    end
    plot(Vmax2(k,:));
    xlabel('Time of Second Pulse Relative to End of First (ms)');
    ylabel('Peak Voltage (mV)');
    hold on;
end
%% Part 6

figure;
thresh = 0;
Irange3 = 1:1:45;
spikes = cell(1,length(Irange3));
ind = spikes;
ISI = spikes;
ISIavg = zeros(1,length(Irange3));
freq = ISIavg;
for k = 1:length(Irange3)
    I = @(t) (t>=1)*Irange3(k);
    [T,Y] = ode45(@(t,in) HH(t,in,I), tspan, y0);
    V = Y(:,1);
    plot(T,V);
    hold on;
    ind{k} = find(V>thresh);
    if isempty(ind{k})
        freq(k) = 0;
        continue;
    end
    inds = find(diff(ind{k}) ~= 1);
    spikes{k} = [ind{k}(1);ind{k}(inds+1)];
    line([T(spikes{k}(1)) T(spikes{k}(1))],[-60 60]);
    if length(spikes{k}) <= 1
        freq(k) = 0;
        print('empty')
    else
        for i = 1:length(spikes{k})-1
            ISI{k} = [ISI{k},(spikes{k}(i+1)-spikes{k}(i))];
        end
        ISIavg(k) = tinc*mean(ISI{k});
        freq(k) = 1000/ISIavg(k);
        line([T(spikes{k}(2)) T(spikes{k}(2))],[-60 60]);
    end
    xlabel('Time (ms)');
    ylabel('Voltage (mV)');
    hold off;
    if k>46
        pause;
    end
end

figure;
plot(Irange3,freq);
xlabel('Injected Current \mu A/cm^2');
ylabel('Firing Frequency (Hz)');
```

The following function is used to evaluate the derivatives of the slow/fast simplified HH model.

```
function [Output] = HHSimp(t,in,I)
C = 0.8;
```

```
E = in(1);
n = in(2);
h = C-n;
Output = zeros(2,1);

G_Na = 120;
G_K = 36;
G_L = 0.3;
E_Na = 55;
E_K = -72;
E_L = -50;

am = @(x) -0.1*(40+x)/(exp(-(40+x)/10)-1);
bm = @(x) 4*exp(-(65+x)/18);

an = @(x) -0.01*(55+x)/(exp(-(55+x)/10)-1);
bn = @(x) 0.125*exp(-(x+65)/80);

Output(2) = an(E)*(1-n)-bn(E)*n;

minf=am(E)/(am(E)+bm(E));
tn = 1/(an(E)+bn(E));
ninf = an(E)/(an(E)+bn(E));

Output(1) = -G_Na.*minf^3.*h.*(E-E_Na)-G_K.*n^4.*(E-E_K)-G_L.*(E-E_L)+I(t);
```

This script performs evaluations of the simplified model in response to a current pulse of amplitude 10 (part 1), plots the phase planes of both the simplified and full models, evaluates the phase plane vectors and finds the nullclines, and plots trajectories for amplitudes 0, 4, and 8. Note that the quiver output was found to scale the derivatives poorly, so scaling factors were manually included along with some changes in the graphics (i.e. the arrowheads were found to be too large and could not be properly scaled, so a triangular marker was placed at the base and the vector magnitudes were inverted so the arrows appear to be at the correct end.

```
E = -61.2;
m0=0.0820;
h0=0.4603;
n0=0.3772;
tinc=0.001;
tspan=0:tinc:30;
c = 0.8;

G_Na = 120;
G_K = 36;
G_L = 0.3;
E_Na = 55;
E_K = -72;
E_L = -50;

%% Part 1

figure;
y0 = [E n0];
I = @(t) (t>=5 && t<6)*10;
[T,Y2] = ode45(@(t,in) HHSimp(t,in,I), tspan, y0);

subplot(3,1,1);
plot(T,Y2(:,1));
xlabel('Time (ms)');
```

```
ylabel('Membrane Voltage (mV)');

am = @(x) -0.1*(40+x)/(exp(-(40+x)/10)-1);
bm = @(x) 4*exp(-(65+x)/18);

for i = 1:length(T)
    Ei = Y2(i,1);
    minf=am(Ei)/(am(Ei)+bm(Ei));
    g_Na(i) = G_Na*minf^3*(c-Y2(i,2));
    g_K2(i) = G_K*Y2(i,2)^4;
    IT(i)=I(T(i));
end

subplot(3,1,2);
plot(T,g_Na);
xlabel('Time (ms)');
ylabel('Sodium Conductance (mS/cm^2)');

subplot(3,1,3);
plot(T,g_K2);
xlabel('Time (ms)');
ylabel('Potassium Conductance (mS/cm^2)');

if exist('comp') && comp == 1
    figure;
    subplot(1,2,1);
    plot(Y(:,1),g_K);
    title('Full Model');
    xlabel('Voltage (mV)');
    ylabel('Potassium Conductance (mS/cm^2)');
    subplot(1,2,2);
    plot(Y2(:,1),g_K2);
    title('Simplified Model');
    xlabel('Voltage (mV)');
    ylabel('Potassium Conductance (mS/cm^2)');
end

%% Part 2
Vrange = -70:2:60;
Vrange = Vrange-1.5;
nrange = 0:0.05:1;
dVrange = zeros(length(Vrange),length(nrange));
dnrange = dVrange;
ninf = zeros(length(Vrange),1);
minf = zeros(length(Vrange),1);
n0 = minf;

an = @(x) -0.01*(55+x)/(exp(-(55+x)/10)-1);
bn = @(x) 0.125*exp(-(x+65)/80);
am = @(x) -0.1*(40+x)/(exp(-(40+x)/10)-1);
bm = @(x) 4*exp(-(65+x)/18);

for i = 1:length(Vrange)
    for j = 1:length(nrange)
        Out = HHSimp(1,[Vrange(i) nrange(j)],0);
        dVrange(i,j) = Out(1);
        dnrange(i,j) = Out(2);
```

```
    end
    Ei = Vrange(i);
    ninf(i) = an(Ei)/(an(Ei)+bn(Ei));
    minf(i) = am(Ei)/(am(Ei)+bm(Ei));
    A = G_K*(Ei-E_K);
    B = G_Na*minf(i)^3*(Ei-E_Na);
    C = G_L*(Ei-E_L)+c*G_Na*minf(i)^3*(Ei-E_Na);
    nroot = roots([A 0 0 -B C]);
    %keyboard;
    re = find(imag(nroot)==0);
    if ~isempty(re)
        n0(i) = nroot(re(end));
    end

end

figure;
%quiver(Vrange,nrange,dVrange',dnrange','AutoScaleFactor',2,'MaxHeadSize
    ',0.0005);
q=quiver(Vrange,nrange,-1*dVrange',-1000.*dnrange');
q.ShowArrowHead = 'off';
q.Marker = '^';
q.MarkerSize = 2;
q.MarkerFaceColor = 'b';
title('2D Phase Plot for Simplified HH Model');
xlabel('Voltage (mV)');
ylabel('Fraction of K Gates Open, n');
hold on;

plot(Vrange,ninf,'r');
plot(Vrange,n0,'b');

%% Part 3
Iin = [0 4 8];
color = ['y','g','k'];
for i = 1:length(Iin)
    I = @(t) (t>=5 && t<6)*Iin(i);
    [T,Y] = ode45(@(t,in) HHSimp(t,in,I), tspan, y0);
    plot(Y(:,1),Y(:,2),color(i));
end
xlim([-75 60]);
ylim([0 1]);
```

This function was used to evaluate the fast-only HH simplified model.

```
function [Output] = HHSimp2(t,in,I)
E = in(1);
m = in(2);
h0 = 0.4603;
n0 = 0.3772;
Output = zeros(2,1);

G_Na = 120;
G_K = 36;
G_L = 0.3;
E_Na = 55;
E_K = -72;
E_L = -50;
```

```
am = @(x) -0.1*(40+x)/(exp(-(40+x)/10)-1);
bm = @(x) 4*exp(-(65+x)/18);

Output(2) = am(E)*(1-m)-bm(E)*m;

tm = 1/(am(E)+bm(E));
minf=am(E)/(am(E)+bm(E));

Output(1) = -G_Na.*m^3.*h0.*(E-E_Na)-G_K.*n0^4.*(E-E_K)-G_L.*(E-E_L)+I(t);
```

Finally, this script performs the same sets of evaluations for the second simplified model as the first, and uses the same considerations in modifying the quiver plot output.

```
E = -61.2;
m0=0.0820;
%E = -65;
%m0 = 0.05;
h0=0.4603;
n0=0.3772;
tinc=0.001;
tspan=0:tinc:30;

G_Na = 120;
G_K = 36;
G_L = 0.3;
E_Na = 55;
E_K = -72;
E_L = -50;

%% Part 1
figure;
y0 = [E m0];
I = @(t) (t>=5 && t<6)*10;
[T,Y3] = ode45(@(t,in) HHSimp2(t,in,I), tspan, y0);

subplot(3,1,1);
plot(T,Y3(:,1));
xlabel('Time (ms)');
ylabel('Membrane Voltage (mV)');

am = @(x) -0.1*(40+x)/(exp(-(40+x)/10)-1);
bm = @(x) 4*exp(-(65+x)/18);

for i = 1:length(T)
    Ei = Y3(i,1);
    g_K3(i) = G_K*n0^4;
    g_Na3(i) = G_Na*h0*Y3(i,2);
    IT(i)=I(T(i));
end

subplot(3,1,2);
plot(T,g_Na3);
xlabel('Time (ms)');
ylabel('Sodium Conductance (mS/cm^2)');

subplot(3,1,3);
plot(T,g_K3);
```

```
xlabel('Time (ms)');
ylabel('Potassium Conductance (mS/cm^2)');

if exist('comp') && comp == 2
    figure;
    subplot(1,2,1);
    plot(Y(:,1),g_Na);
    title('Full Model');
    xlabel('Voltage (mV)');
    ylabel('Sodium Conductance (mS/cm^2)');
    subplot(1,2,2);
    plot(Y3(:,1),g_Na3);
    title('Simplified Model');
    xlabel('Voltage (mV)');
    ylabel('Sodium Conductance (mS/cm^2)');
end

%% Part 2
Vrange = -70:2:60;
Vrange = Vrange-1.5;
mrange = 0:0.05:1;
dVrange = zeros(length(Vrange),length(mrange));
dmrange = dVrange;
minf = zeros(length(Vrange),1);
m0 = minf;

am = @(x) -0.1*(40+x)/(exp(-(40+x)/10)-1);
bm = @(x) 4*exp(-(65+x)/18);

for i = 1:length(Vrange)
    for j = 1:length(mrange)
        Out = HHSimp2(1,[Vrange(i) mrange(j)],0);
        dVrange(i,j) = Out(1);
        dmrange(i,j) = Out(2);
    end
    Ei = Vrange(i);
    minf(i) = am(Ei)/(am(Ei)+bm(Ei));
    B = G_Na*(Ei-E_Na)*h0;
    C = G_L*(Ei-E_L)+G_K*(Ei-E_K)*n0^4;
    mroot = roots([B 0 0 C]);
    %keyboard;
    re = find(imag(mroot)==0);
    if ~isempty(re)
        m0(i) = mroot(re(end));
    end

end

figure;
%quiver(Vrange,mrange,dVrange',dmrange','AutoScaleFactor',2,'MaxHeadSize
    ',0.0005);
q=quiver(Vrange,mrange,-1*dVrange',-25.*dmrange');
q.ShowArrowHead = 'off';
q.Marker = '^';
q.MarkerSize = 2;
q.MarkerFaceColor = 'b';
title('2D Phase Plot for Simplified HH Model');
```

```
xlabel('Voltage (mV)');
ylabel('Fraction of m Gates Open');
hold on;

plot(Vrange,minf,'r');
plot(Vrange,m0,'b');

%% Part 3
Iin = [0 4 8];
color = ['y','g','k'];
for i = 1:length(Iin)
    I = @(t) (t>=5 && t<6)*Iin(i);
    [T,Y] = ode45(@(t,in) HHSimp2(t,in,I), tspan, y0);
    plot(Y(:,1),Y(:,2),color(i));
end
xlim([-75 60]);
ylim([0 1]);
```