CS F441: SEL.TOPICS IN CS

Programming Assignment

Shashank Pandey, 2020A7PS0238H

Approach:

Key assumptions made:

- The conductances and battery voltages of Na+, K+ and the leak current (mostly consisting of Cl- ions) is taken to be as follows:
  gNa0 = 120 mS/cm^2
  ENa = 115 mV
  gK0 = 36 mS/cm^2
  EK = -12 mV
  gL0 = 0.3 mS/cm^2
  EL = 10.6 mV

- The Hodgkin-Huxley gating variables of h (which refers to the inactivation variable of the Na+ channel), m (which refers to the activation variable of the Na+ channel), and n (the activation variable of the K+ channel), and the initial voltage are assumed below.

- The values are as follows:
  m = 0.05, h = 0.54, and n = 0.34; V= -70 mV

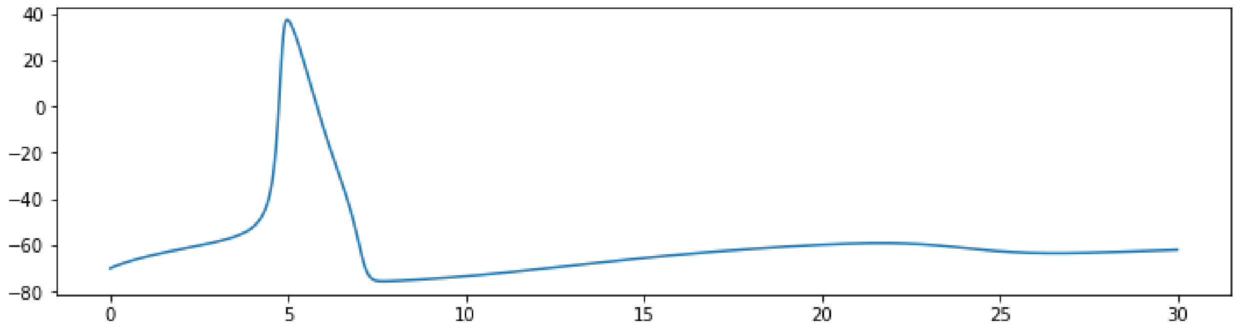
- The Hodgkin-Huxley functions for updation of m,n,h and V were updated by inserting a new value of each variable in an array containing all values of the variables, using its' previously-indexed value in the calculation.

- The rheobase (or the limiting value of current at which a spike occurs) was found experimentally, via testing different approaches- another method to find it could be to iterate through a range of currents, and find the first value of the current parameter where the resultant voltage array contained any positive value.

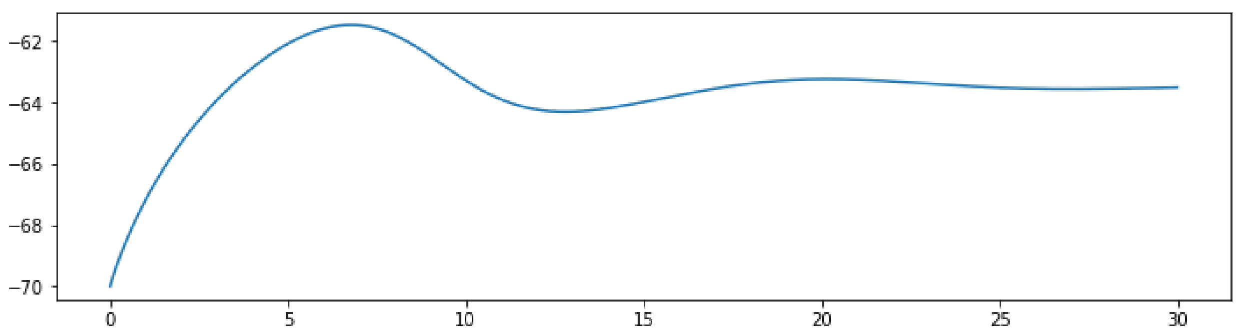
- The firing rate was discretely found, i.e, for every value of input current, the code-defined HodgkHux function calculated the no. of times voltage crossed 0 (as when the voltage crosses 0, it indicates a spike is happening) and cumulatively used that to find the number of spikes.

- Plotting was done via the matplotlib library, and apart from that, the numpy library and its' functions were used for arrays.

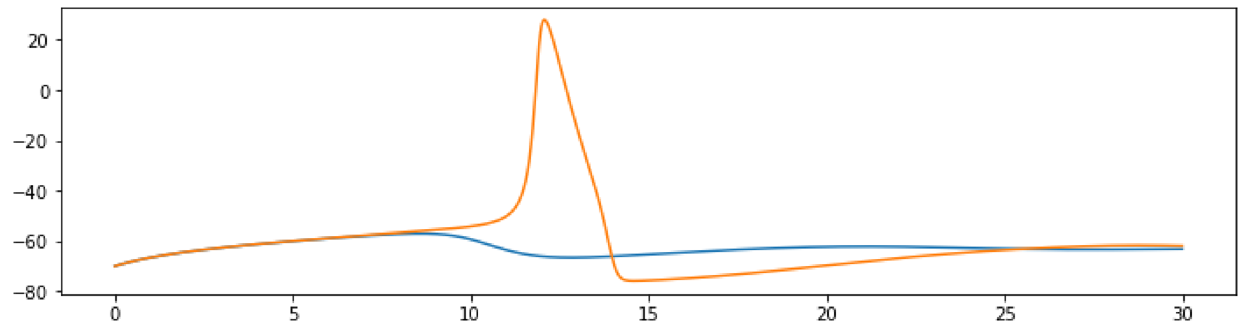
Plots:



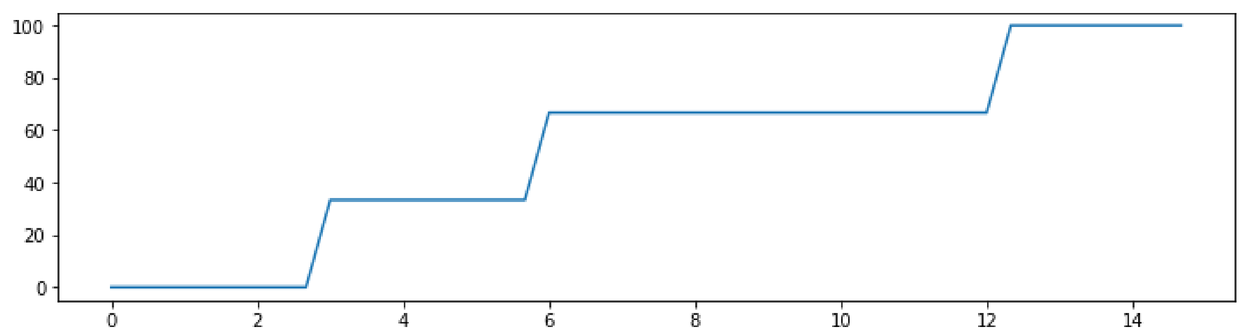
Voltage(Y) vs time(X) for suprathreshold current:
(You can see the potential spike as the entire polarization cycle is followed)
Here, we take input I = 5 mA.



Voltage(Y) vs time(X) for subthreshold current:
(You can see that there is no spike observed, with voltage largely staying the same)
Here, we take input I = 2 mA.



Comparing Voltage(Y) vs time(X) at I=2.89 mA, and at I=2.91 mA. We can experimentally observe that the rheobase current appears to roughly be 2.9 mA.



Finding the firing rate(Y) vs time(X). Since we experimentally count how many times firing has occurred, we get discrete values in our graph, particularly over small amounts of time

Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""

@author: shashankpandey

"""
import numpy as np
import math
from matplotlib.pylab import *


def HodgkHux(I0,T0):
   dt = 0.01; # differential time(suitably small)
   T  = math.ceil(T0/dt)
   gNa0 = 120   # [mS/cm^2]
   ENa  = 115;  # [mV]
   gK0  = 36;   # [mS/cm^2]
   EK   = -12;  # [mV]
   gL0  = 0.3;  # [mS/cm^2]
   EL   = 10.6; # [mV]

   t = np.arange(0,T)*dt
   V = np.zeros([T,1])
   m = np.zeros([T,1])
   h = np.zeros([T,1])
   n = np.zeros([T,1])

   V[0]=-70.0 # initial voltage
   m[0]=0.05 # assumptions
   h[0]=0.54 # of the
   n[0]=0.34 # hh parameters' initial values
   spikes=0

   for i in range(0,T-1):
       V[i+1] = V[i] + dt*(gNa0*m[i]**3*h[i]*(ENa-(V[i]+65)) +
gK0*n[i]**4*(EK-(V[i]+65)) + gL0*(EL-(V[i]+65)) + I0);
       m[i+1] = m[i] + dt*((2.5-0.1*(V[i]+65)) / (np.exp(2.5-0.1*(V[i]+65))
-1)*(1-m[i]) - 4*np.exp(-(V[i]+65)/18)*m[i]);
```

```
        h[i+1] = h[i] + dt*(0.07*np.exp(-(V[i]+65)/20)*(1-h[i]) -
1/(np.exp(3.0-0.1*(V[i]+65))+1)*h[i]);
        n[i+1] = n[i] + dt*((0.1-0.01*(V[i]+65)) / (np.exp(1-0.1*(V[i]+65))
-1)*(1-n[i]) - 0.125*np.exp(-(V[i]+65)/80)*n[i]);
        if (i!=T-1):
            if (V[i+1]>0 and V[i]<=0):
                spikes+=1
   return V,m,h,n,t,spikes*100/3
T0 = 30
x = np.linspace(0,10,30)
I0 = np.array([1 if math.floor(2*t)%2 == 0 else 0 for t in x])
print(I0)

#q1
'''inp1 = float(input("Enter amplitude of current 1: "))
inp2 = float(input("Enter amplitude of current 2: "))
for i in I0:
   V1,m1,h1,n1,t1,spike=HodgkHux(inp1*i,T0)
   V2,m2,h2,n2,t2,spike=HodgkHux(inp2*i,T0)

plot(t1,V1)
plot(t2,V2)'''

#q2
'''inp1 = 2.89
inp2 = 2.91
for i in I0:
   V1,m1,h1,n1,t1,spike=HodgkHux(inp1*i,T0)
   V2,m2,h2,n2,t2,spike=HodgkHux(inp2*i,T0)

plot(t1,V1)
plot(t2,V2)'''
#q3
'''input= [i/3 for i in range(45)]
spik=[0 for i in range(45)]
for j in range(45):
   for i in I0:
       V,m,h,n,t,spik[j]=HodgkHux(input[j]*i,T0)



plot(input,spik)'''
```