

ODE Simulation: Hodgkin-Huxley Action Potential Model

MATH 7205-Fall 2020

Joshua Galloway



Presentation Outline

1. Introduction

- Hodgkin-Huxley Model Introduction

2. Model

- Electrical Circuit
- System of ODE's
- Coefficient Functions
- Fixed Parameters

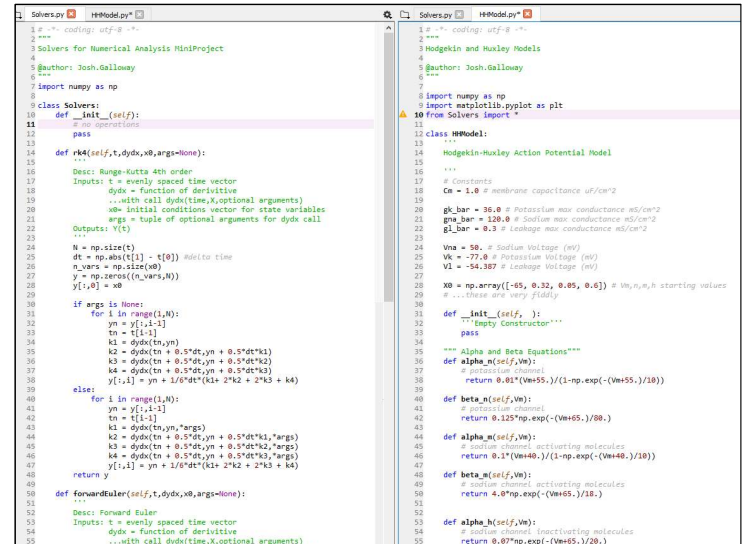
3. Simulation

- Methods Used
- Comparison to Original Paper's Simulation
- Phase Plane

4. Stability and Accuracy

- Stability of Different Methods for Varied Δt
- Error of Different Methods for Varied Δt

5. Conclusion



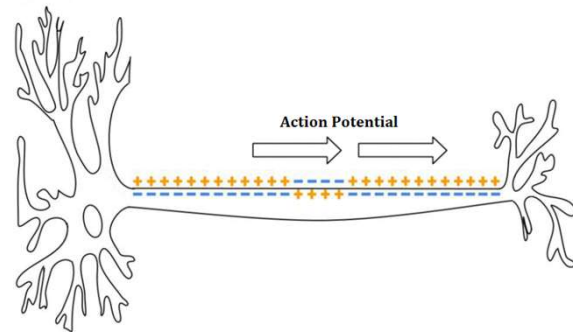
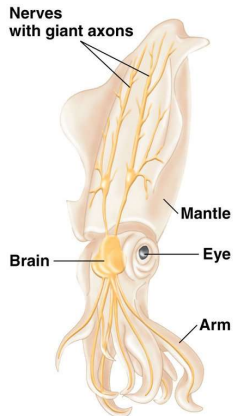
```
1 # coding: utf-8 --
2 ---
3 Solvers for Numerical Analysis MiniProject
4
5 @author: Josh Galloway
6 ---
7 import numpy as np
8
9 class Solvers:
10     def __init__(self):
11         pass
12
13     def rk4(self, t, dydx, x0, args=None):
14         """
15         Desc: Runge-Kutta 4th order
16         Inputs: t = evenly spaced time vector
17                 dydx = function of derivative
18                 ...with call dydx(time, x, optional arguments)
19                 x0 = initial conditions vector for state variables
20                 args = tuple of optional arguments for dydx call
21         Outputs: V(t)
22         """
23         N = np.size(t)
24         dt = np.abs(t[1] - t[0]) #delta t time
25         n_vars = np.size(x0)
26         y = np.zeros((n_vars, N))
27         y[:,0] = x0
28
29         if args is None:
30             for i in range(1,N):
31                 yn = y[:,i-1]
32                 tn = t[i-1]
33                 k1 = dydx(tn, yn)
34                 k2 = dydx(tn + 0.5*dt, yn + 0.5*dt*k1)
35                 k3 = dydx(tn + 0.5*dt, yn + 0.5*dt*k2)
36                 k4 = dydx(tn + 0.5*dt, yn + 0.5*dt*k3)
37                 y[:,i] = yn + 1/6*dt*(k1 + 2*k2 + 2*k3 + k4)
38             else:
39                 for i in range(1,N):
40                     yn = y[:,i-1]
41                     tn = t[i-1]
42                     k1 = dydx(tn, yn, args)
43                     k2 = dydx(tn + 0.5*dt, yn + 0.5*dt*k1, args)
44                     k3 = dydx(tn + 0.5*dt, yn + 0.5*dt*k2, args)
45                     k4 = dydx(tn + 0.5*dt, yn + 0.5*dt*k3, args)
46                     y[:,i] = yn + 1/6*dt*(k1 + 2*k2 + 2*k3 + k4)
47                 return y
48
49     def forwardEuler(self, t, dydx, x0, args=None):
50         """
51         Desc: Forward Euler
52         Inputs: t = evenly spaced time vector
53                 dydx = function of derivative
54                 ...with call dydx(time, x, optional arguments)
55         """
```

```
1 # coding: utf-8 --
2 ---
3 Hodgkin and Huxley Models
4
5 @author: Josh Galloway
6 ---
7
8 import numpy as np
9 import matplotlib.pyplot as plt
10 from Solvers import *
11
12 class HModel:
13     """
14     Hodgkin-Huxley Action Potential Model
15     """
16     # Constants
17     Cn = 1.0 # membrane capacitance uF/cm^2
18     gK_bar = 36.0 # Potassium max conductance nS/cm^2
19     gNa_bar = 120.0 # Sodium max conductance nS/cm^2
20     gL_bar = 0.3 # Leakage max conductance nS/cm^2
21
22     Vna = 50. # Sodium Voltage (mV)
23     Vn = -77.0 # Potassium Voltage (mV)
24     VL = -54.307 # Leakage Voltage (mV)
25
26     x0 = np.array([-65, 0.32, 0.05, 0.6]) # Vm, n, m, h starting values
27     # ...these are very fidely
28
29     def __init__(self, ):
30         """Empty Constructor"""
31         pass
32
33     """Alpha and Beta Equations"""
34     def alpha_m(self, Vm):
35         # potassium channel
36         return 0.87*(Vm55.)/(1-np.exp(-(Vm55.)/10))
37
38     def beta_m(self, Vm):
39         # potassium channel
40         return 0.125*np.exp(-(Vm+65.)/80.)
41
42     def alpha_h(self, Vm):
43         # sodium channel activating molecules
44         return 0.1*(Vm+40.)/(1-np.exp(-(Vm+40.)/10))
45
46     def beta_h(self, Vm):
47         # sodium channel inactivating molecules
48         return 4.0*np.exp(-(Vm+65.)/16.)
49
50     def alpha_n(self, Vm):
51         # sodium channel inactivating molecules
52         return 0.07*np.exp(-(Vm+55.)/28.)
```

Introduction

Hodgkin-Huxley Model:

- Created in 1952 to explain the ionic mechanisms underlying the initiation and propagation of action potentials in the squid giant axon
- Received the Nobel Prize in Physiology and Medicine for this work in 1963
- Electrical model of action potentials in neurons created from experiments on squid giant axons
- Set of 4 non-linear Autonomous ODE's



References: [\[1\]](#), [\[2\]](#), [\[3\]](#)

Model

System of Equations

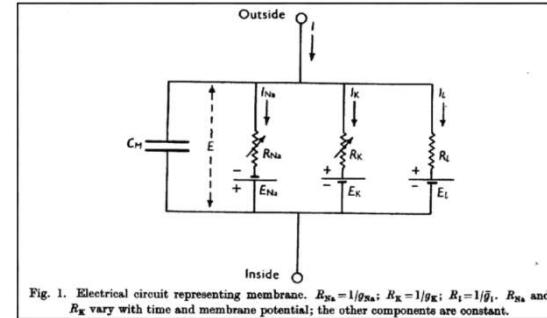
Nonlinear ODE's

$$I = C_m \frac{dV_m}{dt} + \bar{g}_K n^4 (V_m - V_K) + \bar{g}_{Na} m^3 h (V_m - V_{Na}) + \bar{g}_L (V_m - V_L)$$

$$\frac{dn}{dt} = \alpha_n(V_m) \cdot (1 - n) - \beta_n(V_m) \cdot n$$

$$\frac{dm}{dt} = \alpha_m(V_m) \cdot (1 - m) - \beta_m(V_m) \cdot m$$

$$\frac{dh}{dt} = \alpha_h(V_m) \cdot (1 - h) - \beta_h(V_m) \cdot h$$



PDE

$$\frac{a}{2R_2} \frac{\partial^2 V_m}{\partial x^2} = C_m \frac{dV_m}{dt} + \bar{g}_K n^4 (V_m - V_K) + \bar{g}_{Na} m^3 h (V_m - V_{Na}) + \bar{g}_L (V_m - V_L)$$

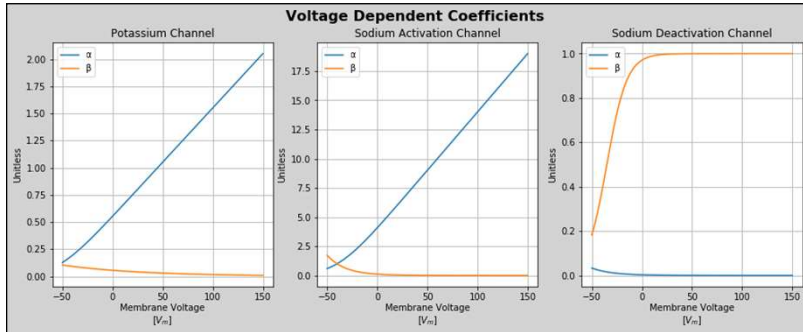
where,

- Na , K , L are sodium, potassium and leakage channel subscripts (i -th ion channel)
- I total current density, I_i channel current density for the i -th ion channel
- C_m membrane capacitance
- V_m voltage across membrane, V_i channel voltage for the i -th ion channel
- α_i and β_i are rate constants for the i -th ion channel
- \bar{g}_i is the maximal value of the conductance for the i -th ion channel
- n , m , and h are dimensionless quantities between 0 and 1 that are associated with potassium channel activation, sodium channel activation, and sodium channel inactivation, respectively
- a axon fibre radius
- R_2 axoplasm specific resistance

Model (continued)

Functions α and β Voltage Dependant Coefficients

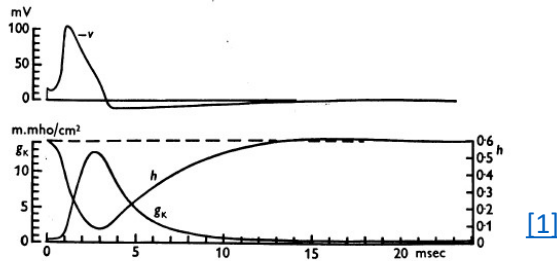
$$\begin{aligned}\alpha_n(V_m) &= \frac{0.01 \cdot (V_m + 55)}{1 - \exp\left(\frac{V_m + 55}{10}\right)} & \beta_n(V_m) &= 0.125 \cdot \exp\left(\frac{-(V_m + 65)}{80}\right) \\ \alpha_m(V_m) &= \frac{0.1 \cdot (V_m + 40)}{1 - \exp\left(\frac{-(V_m + 40)}{10}\right)} & \beta_m(V_m) &= 4 \cdot \exp\left(\frac{-(V_m + 65)}{18}\right) \\ \alpha_h(V_m) &= 0.07 \cdot \exp\left(\frac{-(V_m + 65)}{20}\right) & \beta_h(V_m) &= \left(1 + \exp\left(\frac{-(V_m + 35)}{10}\right)\right)^{-1}\end{aligned}$$



Parameters Used	Capacitance [uF/cm ²]	V_i [mV]	\bar{g}_i [mS/cm ²]	Initial Values
Membrane	1.0	--	--	$V_m = -65$
Potassium Channel	--	-77.0	0.5	$n = 0.32$
Sodium Activation	--	50.0	120.0	$m = 0.05$
Sodium Deactivation	--	50.0	120.0	$h = 0.6$
Leakage Channel	--	-54.387	0.3	--

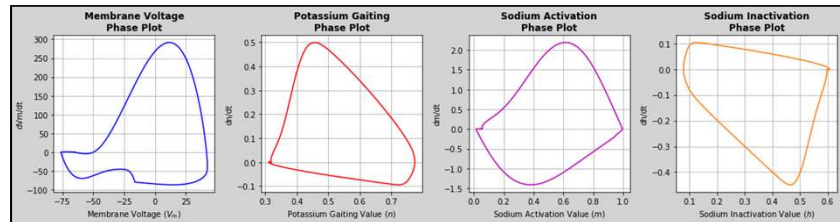
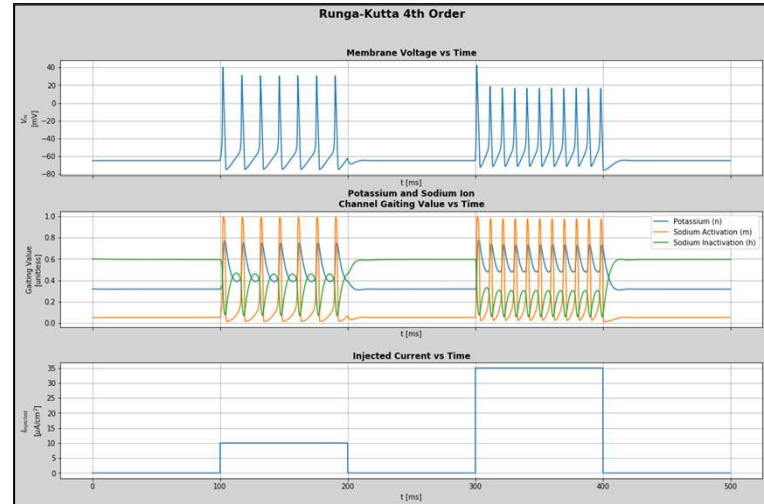
Simulation

- Forward Euler, Huen's Method, and 4th Order Runge-Kutta Implemented from Scratch in Python using Numpy



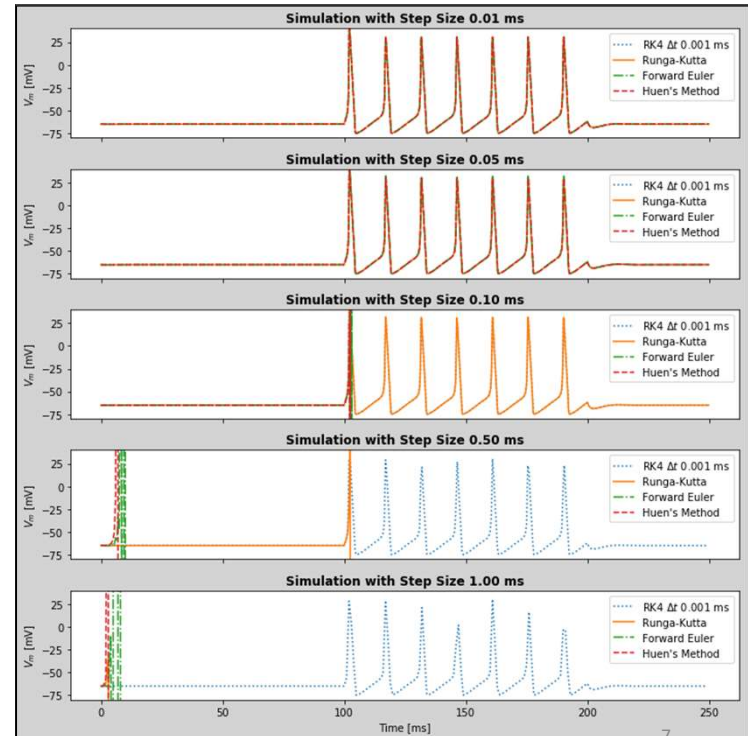
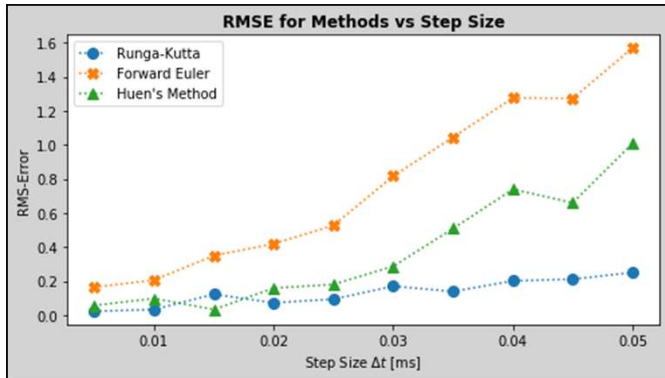
[1]

Fig. 19. Numerical solution of eqn. (26) for initial depolarization of 15 mV and temperature of 6° C. Upper curve: membrane potential, as in Fig. 13. Lower curves show time course of g_K and h during action potential and refractory period.



Stability and Accuracy

- Vary Step Size for Simulation and Observe when the Methods become Unstable
- Compare Each Simulation to a Benchmark of Runga-Kutta 4th Order with a Step Size of 0.001 ms
- Calculate the Root Mean Squared Error for Step Sizes in [0.005,0.05] and Compare Against the Benchmark



Conclusion

We Saw...

Model

- Electrical Circuit, Model Equations

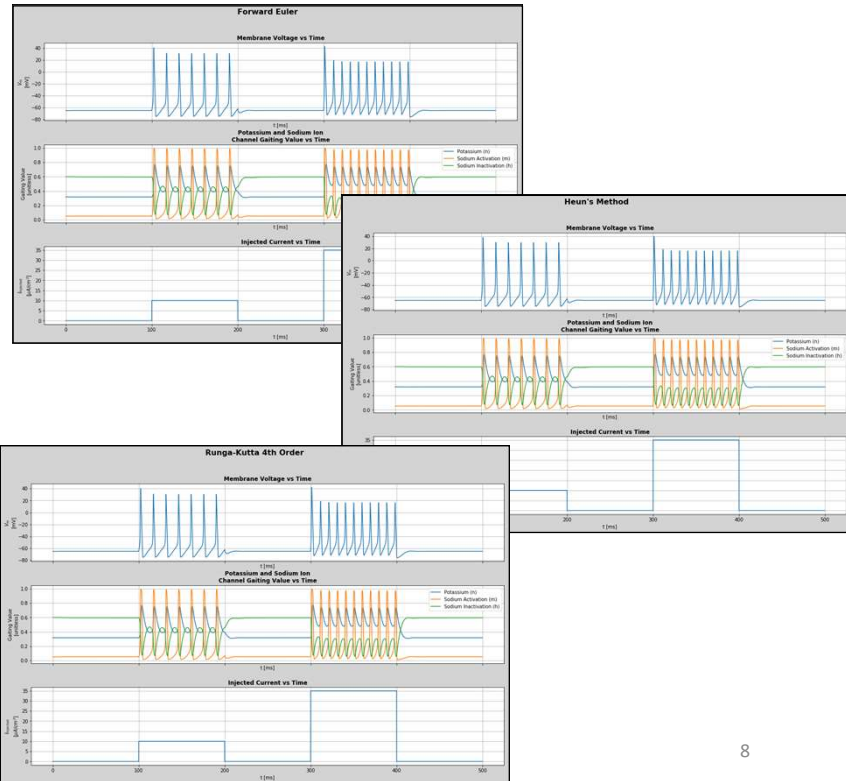
Simulation

- Forward Euler, Huen's Method, Runga-Kutta 4th Order
- Simulations all Matched the Original Paper with Proper Parameter Selection

Stability and Accuracy

- Runga-Kutta Performed Best in both Stability and Accuracy followed by Huen's Method then Forward Euler as would be Expected

Questions?....



References

1. Hodgkin AL and AF Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol., 117:500–544, 1952
2. Hodgkin–Huxley model (2020, May 22). Retrieved Oct 08, 2020, from https://en.wikipedia.org/wiki/Hodgkin-Huxley_model
3. A Computational View of the Historical Controversy on Animal Electricity - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-squid-giant-axon-The-giant-axon-is-a-very-large-up-to-1-mm-in-diameter-and-long_fig2_276491039 [accessed 21 Oct, 2020]