7. Hodgkin-Huxley model coding

Taehoon Kim

Contents

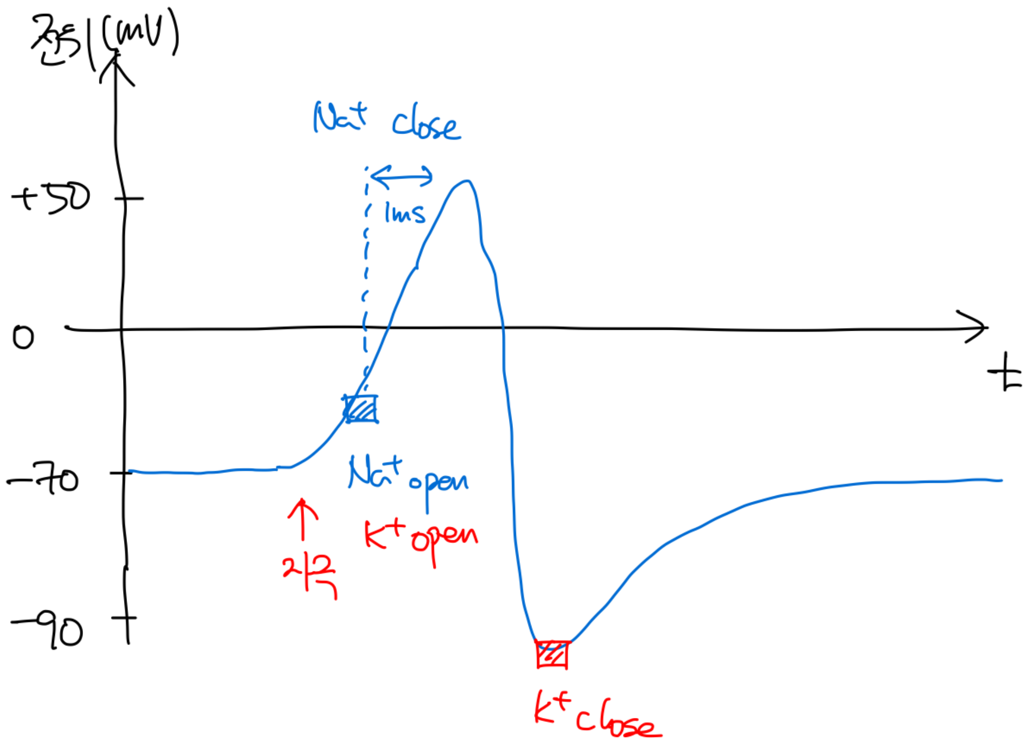
Review

Biological Backgounds

-55mV: Voltage-gated Na+, K+ channel open

After 1ms: Na+ channel closed (Activation gate)

-90mV: Voltage-gated K+ channel close



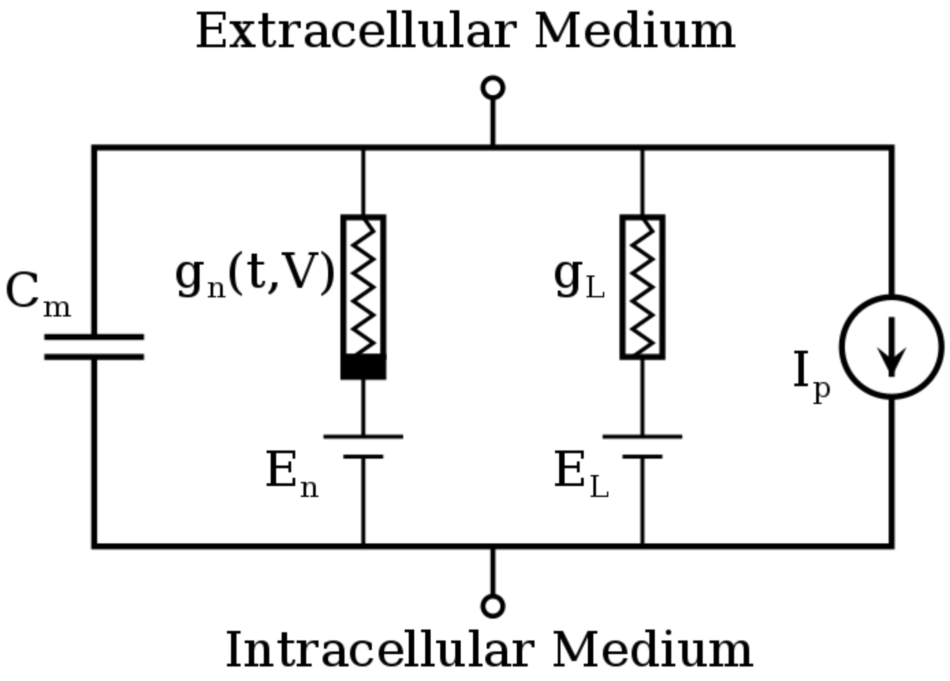
Hodgkin-Huxley model

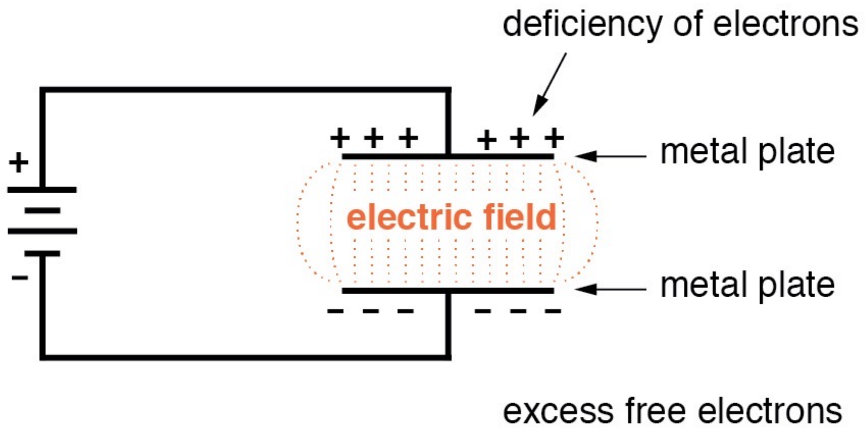
Kirchhoff's Law

$$I_{tot} = \underbrace{I_m}_{Capacitor} + \underbrace{I_{Na} + I_K + I_{Leak}}_{Conductance}$$

Membrane Current (Capacitor)

$$Q = C_m V \Rightarrow I_m = \frac{\mathrm{d}Q}{\mathrm{d}t} = C_m \frac{\mathrm{d}V}{\mathrm{d}t}$$





Hodgkin-Huxley model

Ion flow (Conductance)

$$I \propto V \Rightarrow I = gAV$$

$$g_K = \bar{g}_K n^4$$

$$g_{Na} = \bar{g}_{Na} m^3 h$$

Voltage-gated Ion channel

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(V)(1-n) - \beta_n(V)n$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(V)(1-m) - \beta_m(V)m$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(V)(1-h) - \beta_H(V)h$$

$$\alpha_n = \frac{0.01(V+55)}{1-\exp(-0.1(V+55))}, \qquad \beta_n = 0.125\exp(-(V+65)/80)$$

$$\alpha_m = \frac{0.1(V+40)}{1-\exp(-0.1(V+40))}, \qquad \beta_m = 4\exp(-(V+65)/18)$$

$$\alpha_h = 0.07\exp((V+65)/20), \qquad \beta_n = \frac{1}{1+\exp(-0.1(V+35))}$$

Hodgkin-Huxley model

Final solution (Hodgkin-Huxley model)

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

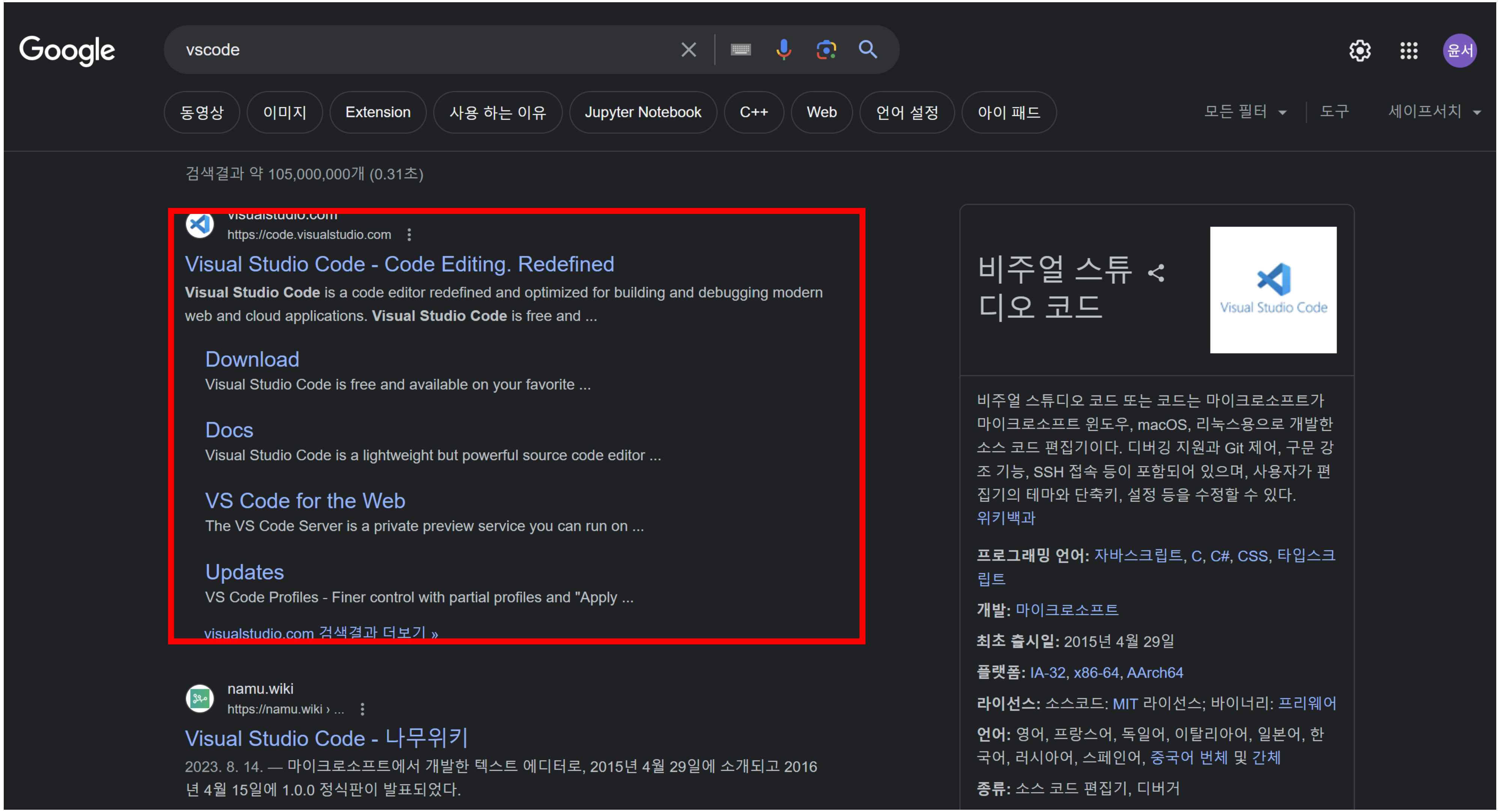
$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(V)(1 - n) - \beta_n(V) n$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(V)(1 - m) - \beta_m(V) m$$

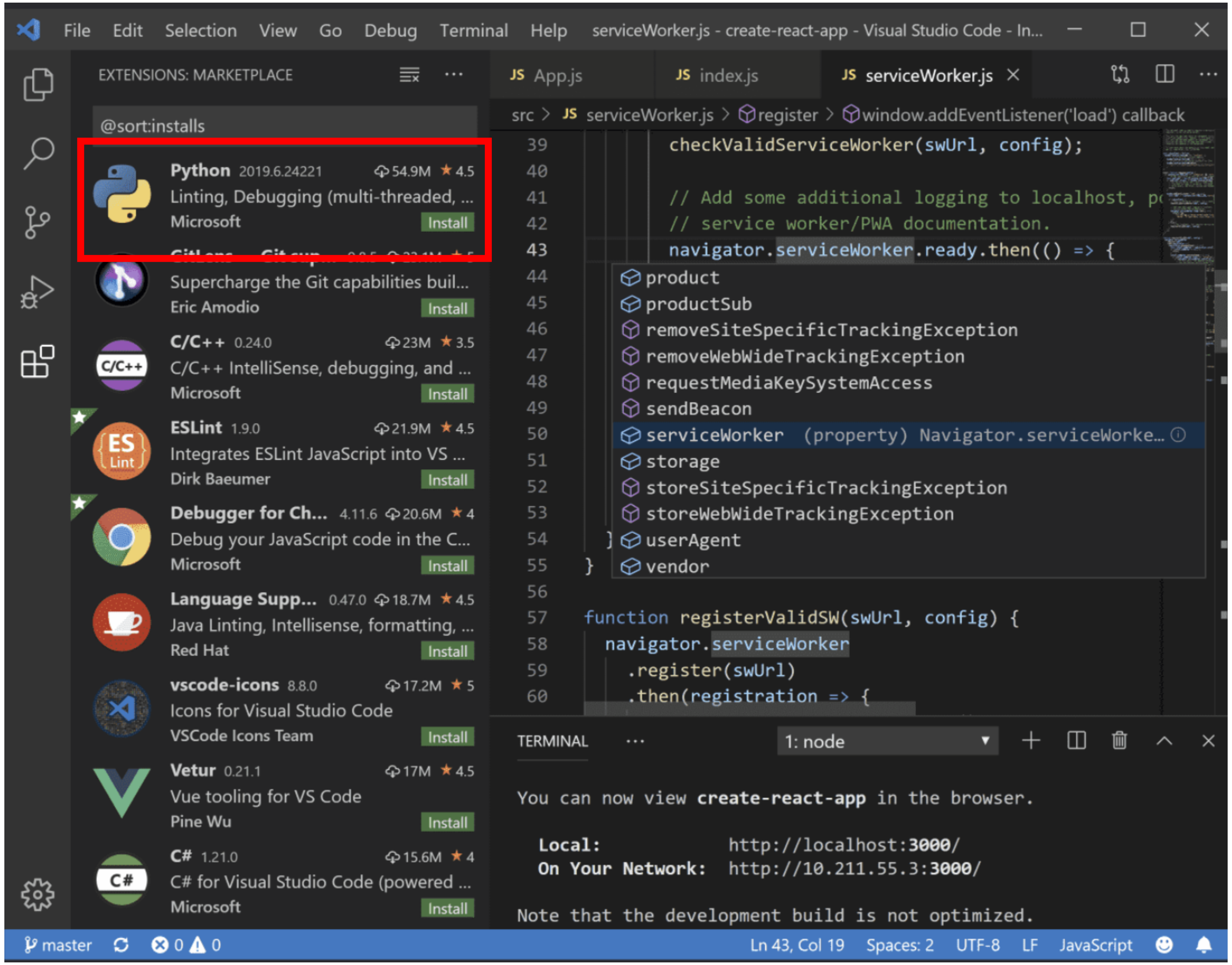
$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(V)(1 - h) - \beta_H(V) h$$

Hodgkin-Huxley model coding

Download VS code



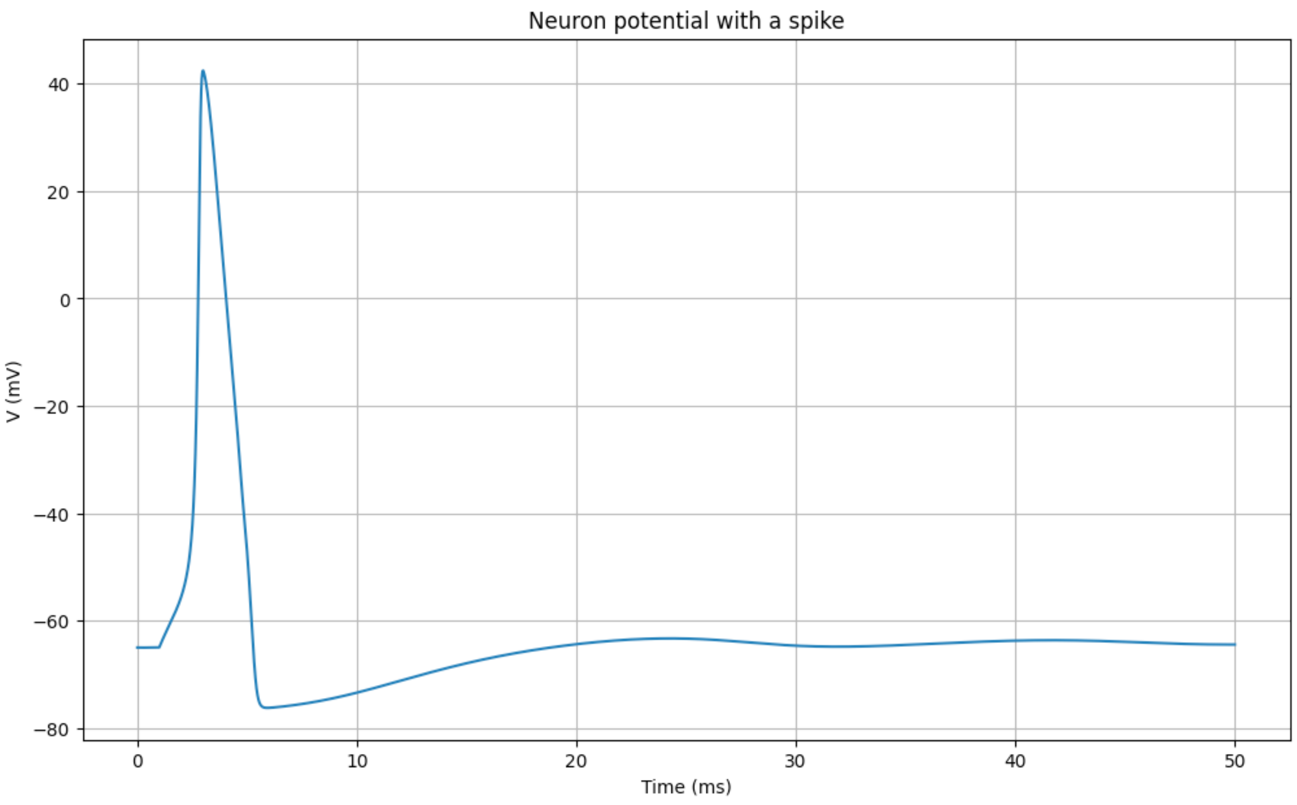
Download VS code



Hodgkin-Huxley model coding

Our Goal:

implementing action potential with Hodgkin-Huxley model!



Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

STEP2. Setting Parameters

STEP3. Solving ODE

STEP4. Plotting Results

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

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

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(V)(1-n) - \beta_n(V)n \qquad \alpha_n = \frac{0.01(V+55)}{1-\exp(-0.1(V+55))}, \qquad \beta_n = 0.125\exp(-(V+65)/80)$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(V)(1-m) - \beta_m(V)m \qquad \alpha_m = \frac{0.1(V+40)}{1-\exp(-0.1(V+40))}, \qquad \beta_m = 4\exp(-(V+65)/18)$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(V)(1-h) - \beta_H(V)h \qquad \alpha_h = 0.07\exp((V+65)/20), \qquad \beta_n = \frac{1}{1+\exp(-0.1(V+35))}$$

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

```
# data ref: theoretical Neuroscience
# source ref: https://www.bonaccorso.eu/2017/08/19/hodgkin-huxley-spiking-neuron-
model-python/

import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint
```

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

```python
# Potassium ion-channel rate functions
def alpha_n(V):
     return (0.01 * (V+55)) / (1- np.exp(-(V+55)/10))
def beta_n(V):
     return 0.125 * np.exp(-(V+65) / 80.0)

# Sodium ion-channel rate functions
def alpha_m(V):
     return (0.1 * (V + 40)) / (1-np.exp(-(V+40)/10))
def beta_m(V):
     return 4.0 * np.exp(-(V+65) / 18.0)
def alpha_h(V):
     return 0.07 * np.exp(-(V +65)/ 20.0)
def beta_h(V):
     return 1.0 / (1 + np.exp(-(V+35)))
```

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

```
# Compute derivatives
def compute_derivatives(y, t0):
    dy = np.zeros((4,))
    V = y[0]
    n = y[1]
    m = y[2]
    h = y[3]
    # dV/dt
    GK = (gK / Cm) * np.power(n, 4.0)
    GNa = (gNa / Cm) * np.power(m, 3.0) * h
    GL = gL / Cm
    dy[0] = (Id(t0) / Cm) - (GK * (V - VK)) - (GNa * (V - VNa)) - (GL * (V - Vl))
    # dn/dt
    dy[1] = (alpha_n(V) * (1.0 - n)) - (beta_n(V) * n)
    # dm/dt
    dy[2] = (alpha_m(V) * (1.0 - m)) - (beta_m(V) * m)
    # dh/dt
    dy[3] = (alpha_h(V) * (1.0 - h)) - (beta_h(V) * h)
    return dy
```

Hodgkin-Huxley model coding

STEP2. Setting Parameters

```
# Average potassium channel conductance per unit area (mS/cm^2)
gK = 36.0
# Average sodoum channel conductance per unit area (mS/cm^2)
gNa = 120.0
# Average leak channel conductance per unit area (mS/cm^2)
gL = 0.3
# Membrane capacitance per unit area (uF/cm^2)
Cm = 1.0
# Potassium potential (mV)
VK = -77
# Sodium potential (mV)
VNa = 50
# Leak potential (mV)
Vl = -54.387
```

Hodgkin-Huxley model coding

STEP2. Setting Parameters

```
# Time values
# Start and end time (in milliseconds)
tmin = 0.0
tmax = 50.0
T = np.linspace(tmin, tmax, 10000)

# Input stimulus
def Id(t):
     if 1.0 < t < 3.0:
          return 10.0
     return 0.0
```

STEP3. Setting Parameters

```
# Initial Condition (V, n, m, h)
Y = np.array([-65, 0.317, 0.05, 0.6])

# Solve ODE system
Vy = odeint(compute_derivatives, Y, T)
```

Hodgkin-Huxley model coding

STEP4. Plotting Results

```
# Input stimulus
Idv = [Id(t) for t in T]
fig, ax = plt.subplots(figsize=(12, 7))
ax.plot(T, Idv)
ax.set_xlabel('Time (ms)')
ax.set_ylabel(r'Current density (uA/$cm^2$)')
ax.set_title('Stimulus (Current density)')
plt.grid()


# Neuron potential
fig, ax = plt.subplots(figsize=(12, 7))
ax.plot(T, Vy[:, 0])
ax.set_xlabel('Time (ms)')
ax.set_ylabel('V (mV)')
ax.set_title('Neuron potential with a spike')
plt.grid()
```

References

- datas & equation ref:

  Theoretical Neuroscience

- source ref:

  https://www.bonaccorso.eu/2017/08/19/hodgkin-huxley-spiking-neuron-model-python/