Hodgkin-Huxley

David Wallace Croft, M.Sc.
Atzori Lab, U.T. Dallas
david@CroftSoft.com

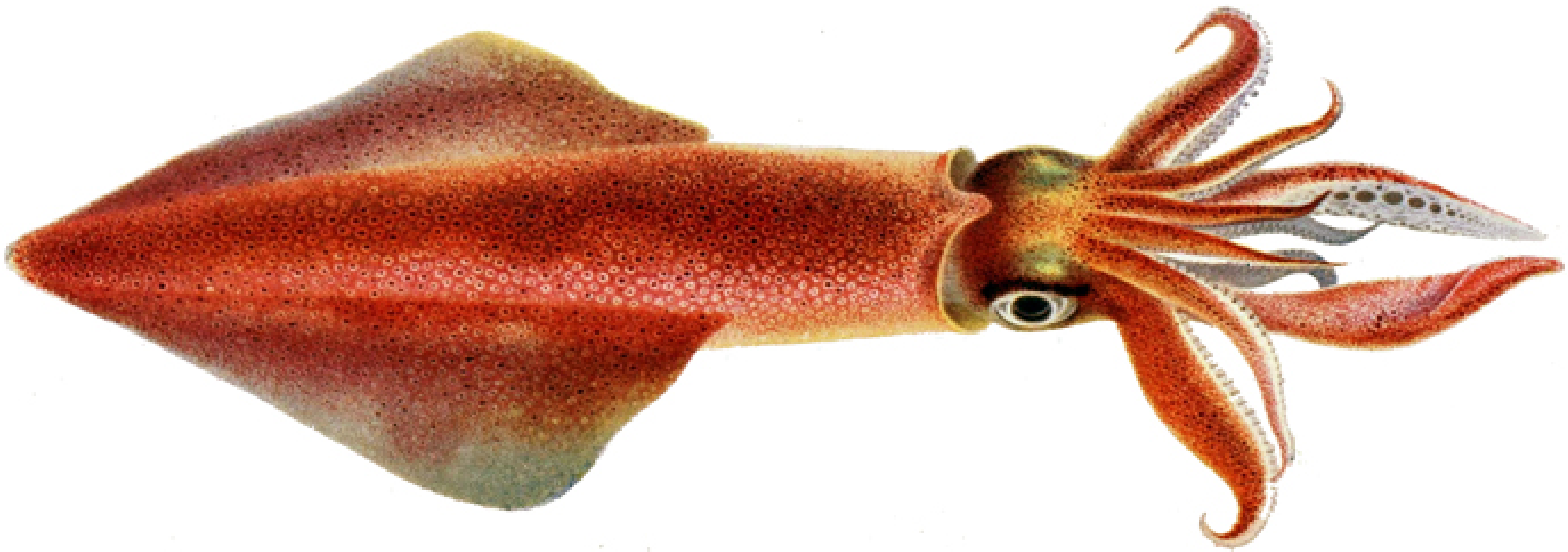
2011 Aug 19 Fri

Overview

- The Hodgkin-Huxley equations provide a biologically realistic model of neural action potentials
- The model is described using differential equations based on laws of physics and experimental measurements
- A computer simulation is demonstrated which uses Euler's Method for numerical integration with a very small time step

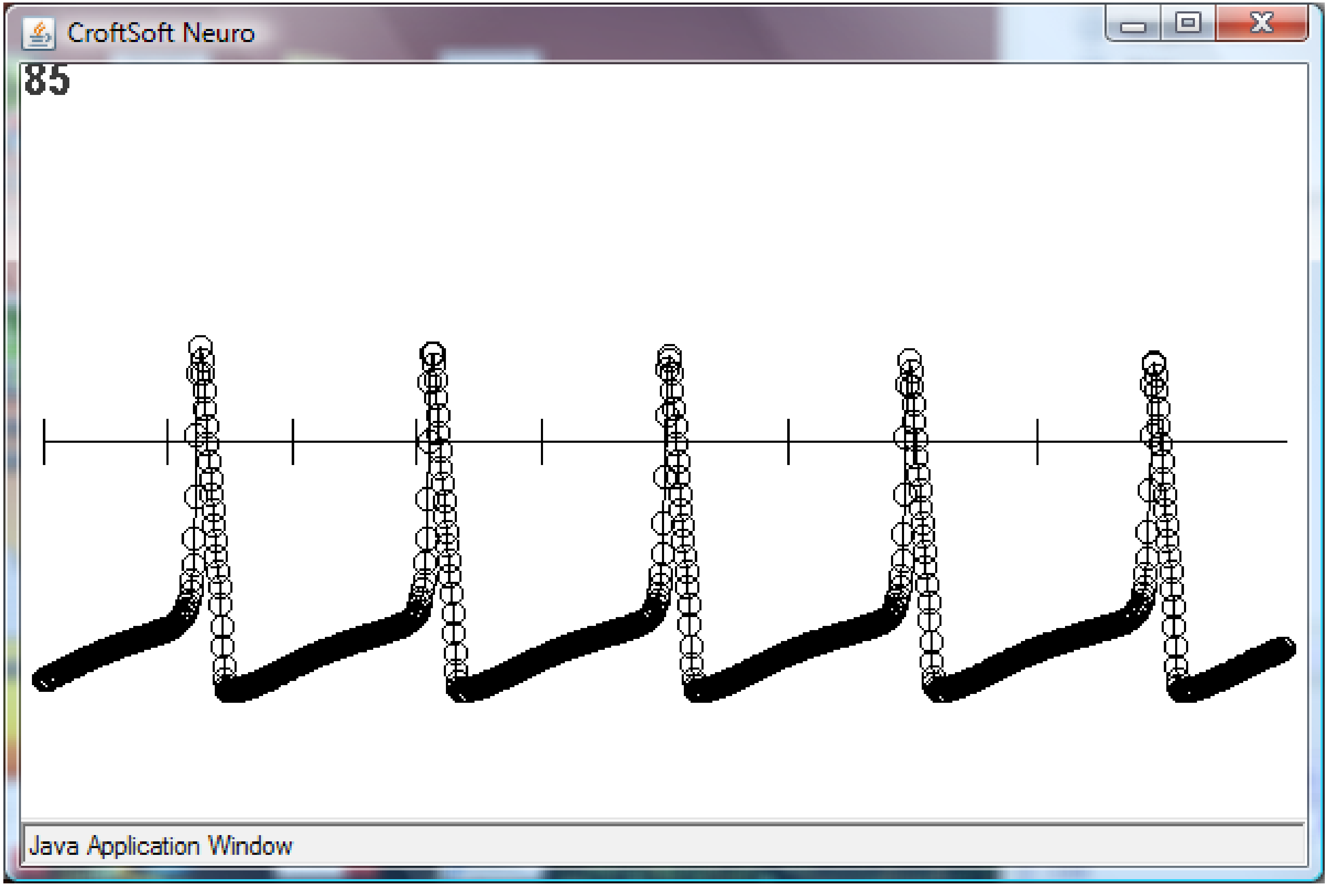
Giant Squid Axon

The axon is giant, not the squid



http://commons.wikimedia.org/wiki/File:Loligo_vulgaris1.jpg

Demonstration

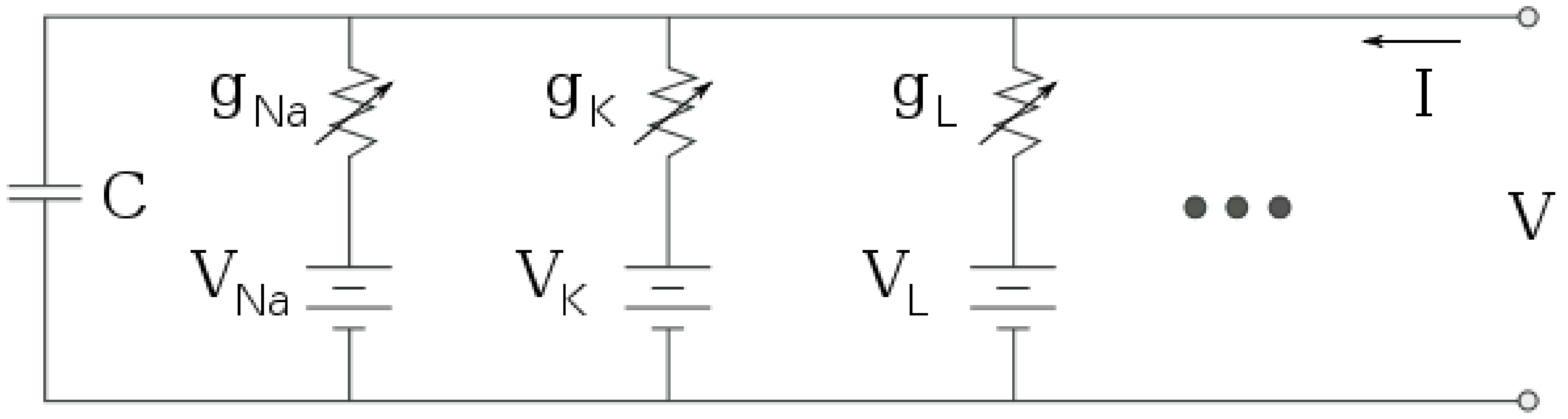


http://www.CroftSoft.com/library/software/neuro/

Demonstration

- Injected current drives action potentials / spikes
- Spiking ceases due to adaptation
- Spiking resumes when injected current is toggled off and then back on
- Intermittent current pulses are less likely to cause an action potential immediately following an action potential during the refractory period
- The membrane voltage changes rapidly between time steps during action potentials

Equivalent Circuit



Arne Nordmann, http://commons.wikimedia.org/wiki/File:Hodgkin-huxley-circuit.svg

Math Notation and Symbols

- Δ Delta: difference; a step-wise change
- x' Prime: first derivative of $x(t)$

Physics Equations

- $c = A \cdot C$
  membrane capacitance = area • capacitivity
- $g = A \cdot G$
  channel conductance = area • conductivity
- $i = g \cdot (v - E)$
  current = conductance • (voltage - reversal potential)
- $\Delta q = \Delta t \cdot (-i)$
  Δ charge = simulation time step • negative current
- $\Delta v = \Delta q / c$
  Δ membrane voltage = Δ charge / capacitance

Hodgkin-Huxley Equations 1

Voltage-dependent parameters

$$\alpha_h = 0.07 \cdot e^{-V/20}$$

$$\beta_h = \frac{1}{e^{(30-V)/10} + 1}$$

$$\alpha_m = \frac{0.1 \cdot (25 - V)}{e^{(25-V)/10} - 1}$$

$$\beta_m = 4 \cdot e^{-V/18}$$

$$\alpha_n = \frac{0.01 \cdot (10 - V)}{e^{(10-V)/10} - 1}$$

$$\beta_n = 0.125 \cdot e^{-V/80}$$

Hodgkin-Huxley Equations 2

$$h_\infty = \alpha_h / (\alpha_h + \beta_h) \qquad \tau_h = 1/(\alpha_h + \beta_h)$$

$$m_\infty = \alpha_m / (\alpha_m + \beta_m) \qquad \tau_m = 1/(\alpha_m + \beta_m)$$

$$n_\infty = \alpha_n / (\alpha_n + \beta_n) \qquad \tau_n = 1/(\alpha_n + \beta_n)$$

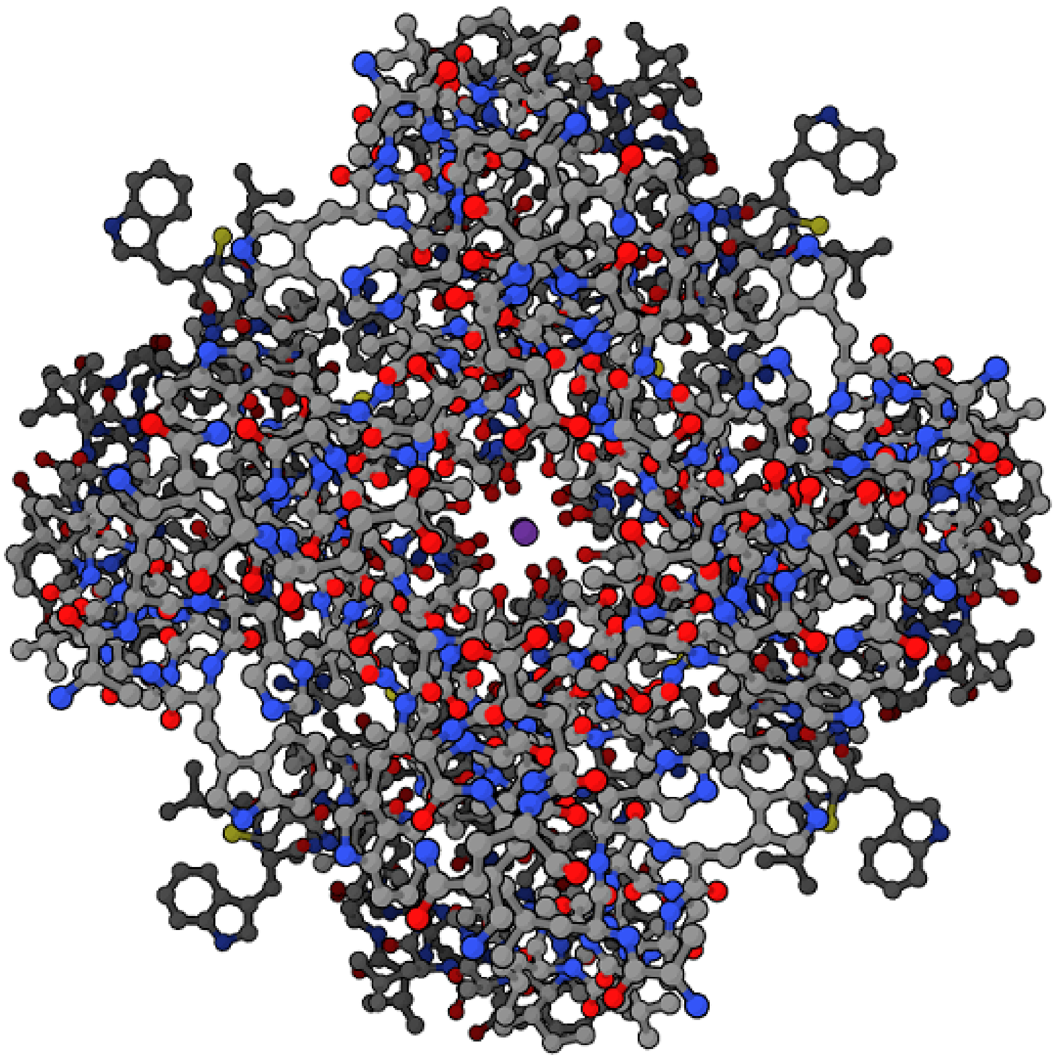
$$h' = (h_\infty - h)/\tau_h$$

$$m' = (m_\infty - m)/\tau_m$$

$$n' = (n_\infty - n)/\tau_n$$

$$C \cdot V' = -g_K \cdot n^4 \cdot (V_m - E_K) - g_{Na} \cdot m^3 \cdot h \cdot (V_m - E_{Na}) - g_L \cdot (V_m - E_L) + I_{synaptic}$$

Potassium Channel



http://en.wikipedia.org/wiki/File:Potassium_channel1.png

Simulation Parameters

```
public static final double
  H_INIT                          =   0.596111046, // steady-state resting
  LEAK_CONDUCTIVITY               =   0.3e+1,      //   0.3 mS/cm^2
  LEAK_REVERSAL_POTENTIAL         = -54.4e-3,      // -54.4 mV
  LHOPITAL_THRESHOLD              =   1e-9,        // L'Hopital's Rule
  M_INIT                          =   0.052934218, // steady-state resting
  MEMBRANE_CAPACITIVITY           =   1e-2,        //   1 microF/cm^2
  MEMBRANE_VOLTAGE_INIT           =  -0.064999722, // steady-state resting
  N_INIT                          =   0.317681168, // steady-state resting
  POTASSIUM_CONDUCTIVITY          =  36e+1,        //  36 mS/cm^2
  POTASSIUM_REVERSAL_POTENTIAL    = -77e-3,        // -77 mV
  SODIUM_CONDUCTIVITY             = 120e+1,        // 120 mS/cm^2
  SODIUM_REVERSAL_POTENTIAL       =  50e-3,        //  50 mV
  THRESHOLD                       =   0;           // crossing for spike
```

Synaptic Currents

```
double  capacitiveCurrent = 0;

for ( int  i = 0; i < size; i++ )
{
  final Channel  channel = channelSeq.get ( i );

  if ( channel.isOpen ( ) )
  {
    final double  channelCurrent = channel.getConductance ( )
      * ( membraneVoltage - channel.getReversalPotential ( ) );

    capacitiveCurrent += channelCurrent;
  }
}
```

Hodgkin-Huxley Currents

```
final double  hhCurrent
  = sodiumConductance * m * m * m * h
    * ( membraneVoltage - sodiumReversalPotential )
  + potassiumConductance * n * n * n * n
    * ( membraneVoltage - potassiumReversalPotential )
  + leakConductance
    * ( membraneVoltage - leakReversalPotential );

capacitiveCurrent += hhCurrent;

final double  deltaCharge = deltaTime * -capacitiveCurrent;

deltaMembraneVoltage = deltaCharge / membraneCapacitance;
```

References

- Croft (2010) "HhNeuronImp.java", http://croftsoft.svn.sourceforge.net/viewvc/croftsoft/[...]
- Croft (2008) "Neuro", http://www.croftsoft.com/library/software/neuro/
- Trappenberg (2010) "Section 2.3 The generation of action potentials: Hodgkin-Huxley equations", Fundamentals of Computational Neuroscience, 2nd Ed., Oxford University Press
- Wikipedia, "Action potential"
- Wikipedia, "Hodgkin-Huxley model"