Cell Simulation Software

Software for Cell Simulation

Results **1** - **10** of about **344,000** for **action potential** **simulation** **software**

1. Propagated **Action Potential**This program simulates the generation and propagation of the **action potential** by solving the Hodgkin and Huxley (HH) model for the space-clamped (membrane **...**
2. nerve.bsd.uchicago.edu/nerve1.html - 7k - Cached - Similar pages -

iROCKET Learning Module: Cardiac **Action Potential Simulator**

The Cardiac **Action Potential Simulator** aims to conduct a graphical review of cardiac membrane physiology and to demonstrate the effects of common **...**

missinglink.ucsf.edu/lm/cardiac_**action_potential**_sim/ - 10k

Simulation of developmental changes in **action potentials** with **...**

Computational Challenges in Cell **Simulation**: A **Software** Engineering Approach · ArticleReference **Simulation** of developmental changes in **action potentials** **...**

www.e-cell.org/ecell/bibliographyfolder/itoh_**simulation**_2007 - 53k

Software for characterizing the ionic basis of the molluscan **...**

simulation software will be useful for developing hypotheses **...** Recordings of **action potentials** in the. molluscan cardiac muscle by use of the single- **...**
ieeexplore.ieee.org/iel5/6792/18228/00842415.pdf?arnumber=842415

Computing Action Potentials

Compute new ion currents

```
[iNa(i),gNa(i),mNa,hNa,jNa] = INa(V(i),th,mNa,hNa,jNa);
[isi(i),gsi(i),msi,hsi] = Isi(V(i),th,msi,hsi);
[iK(i),gK(i),mK] = IK(V(i),th,mK);
[iK1(i),gK1(i),ht,tauht] = IK1(V(i));
[iKp(i),gKp(i)] = IKp(V(i));
[ib(i),gb(i)] = Ib(V(i));
```

Compute total current (Id = stimulus)

```
I = Id+iNa(i)+isi(i)+iK(i)+iK1(i)+iKp(i)+ib(i);
```

Compute incremental voltage change

```
dV = -I/Cm;
```

Update value of voltage and time

```
V(i+1) = V(i) + dV*th;
t(i+1) = t(i) + th;
```

Increment counter

```
i = i+1;
```


Computing the Currents: IKp (plateau)

$$i_{Kp} = \bar{g_{Kp}} K_p (V_m - E_K)$$

$$K_p = \frac{1}{1 + e^{\frac{7.488 - V_m}{5.98}}}$$

```
function [I,gKp] = IKp(V)
% IKp    Plateau potassium current
%   I = IKp(V)
%    I   potassium current
%   gKp  conductance
%    V   membrane potential (mV)
% Reference: QN:Luo94a

global R T F K_o K_i;
GKp_ = 0.0183;
EKp = (R*T/F)*log(K_o/K_i);

numTimeStep = length(V);

for i=1:numTimeStep
  Kp = 1/(1+exp((7.488-V(i))/5.98));
  gKp(i) = GKp_*Kp;
  I(i) = gKp(i)*(V(i)-EKp);
end
```

Computing the Currents: IK (time dependent)

$$i_K = \bar{g_K} m h (V_m - E_K)$$

$$\bar{g_K} = g_K \sqrt{\frac{[K^+]_e}{5.4}}$$

$$h = 1.0 \qquad h = 2.837 \frac{e^{0.04(V_m+77)} - 1}{(V_m + 77)e^{0.04(V_m+35)}}$$

```
function [I,gK,mm] = IK(V,th,m)
 GK_ = 0.282*sqrt(K_o/5.4);          % millisiemens/uF
 EK = (R*T/F)*log((K_o+PNa_K*Na_o)/(K_i+PNa_K*Na_i));        % mV

 if V > -100
     h = 2.837*(exp(0.04*(V+77))-1)/((V+77)*exp(0.04*(V+35)));
 else
   h = 1;
 end

 gK = GK_*m*h;
 I = gK*(V-EK);
```


Computing the Currents: IK activation

$$m = m_\infty(v_o) - (m_\infty(v_o) - m_0)e^{-t/\tau_m(v_o)}$$

$$\tau_m = \frac{1}{\alpha_m(V) + \beta_m(V)} \qquad m_\infty = \frac{\alpha_m(V)}{\alpha_m(V) + \beta_m(V)}$$

$$\alpha_m = 0.005\left(\frac{e^{0.083(V_M+50)}}{1 + e^{0.057(V_M+50)}}\right) \qquad \beta_m = 0.0013\left(\frac{e^{-0.06(V_M+20)}}{1 + e^{-0.04(V_M+20)}}\right)$$

```
am = 0.0005*exp(0.083*(V+50))/(1+exp(0.057*(V+50)));
bm = 0.0013*exp(-0.06*(V+20))/(1+exp(-0.04*(V+20)));

minf = am/(am+bm);
taum = 1/(am+bm);

mm = minf - (minf-m)*exp(-th/taum);
```

Computing the Currents: IK1 (time independent)

$$i_{K1} = \bar{g_{K1}} K_{1\infty}(V_m - E_K)$$

$$\bar{g_{K1}} = g_{K1}\sqrt{\frac{[K+]_e}{5.4}} \qquad K_{1\infty} = \frac{\alpha_{K1}}{\alpha_{K1} + \beta_{K1}}$$

```
function [I,gK1,hinf,tauh] = IK1(V)
GK1_ = 0.6047*sqrt(K_o/5.4);
EK1 = (R*T/F)*log(K_o/K_i);

numTimeStep = length(V);

for i=1:numTimeStep
  ah(i) = 1.02/(1+exp(0.2385*(V(i)-EK1-59.215)));
  bh(i) = (0.49124*exp(0.08032*(V(i)-EK1+5.476)) + ...
      exp(0.06175*(V(i)-EK1-594.31)))/  (1+exp(-0.5143*(V(i)-EK1+4.753)));

  hinf(i) = ah(i)/(ah(i)+bh(i));
  tauh(i) = 1/(ah(i)+bh(i));

  gK1(i) = GK1_*hinf(i);
  I(i) = gK1(i)*(V(i)-EK1);
end
```


Optimization of Code

- Time step

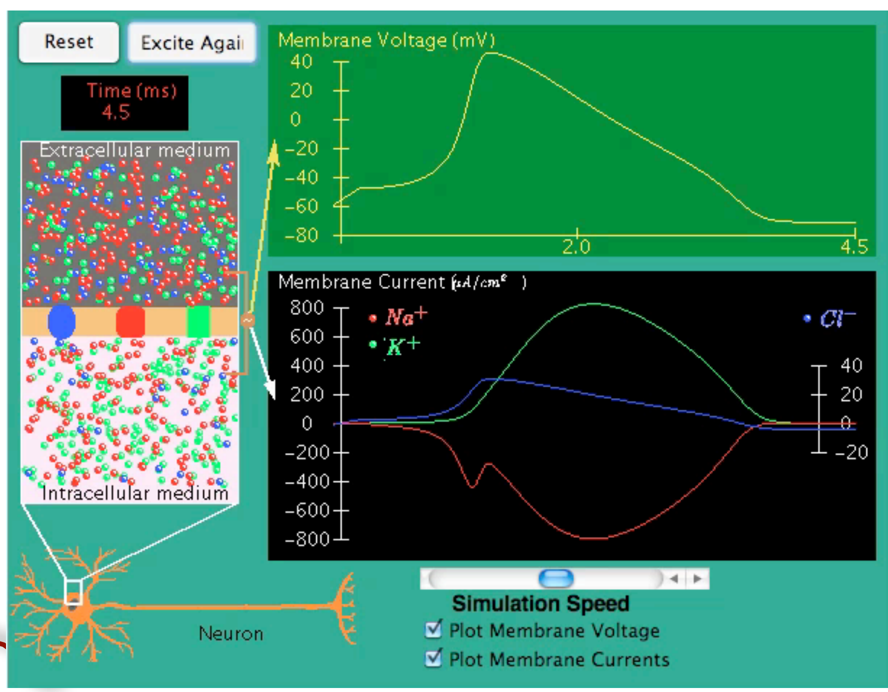
```
I = Id+iNa(i)+isi(i)+iK(i)+iK1(i)+iKp(i+ib(i);
    dV = -I/Cm;
    th = TSmin*dVmax/abs(dV);
    if th > Tsmax
       th = TSmax;
    elseif th < Tsmin
       th = TSmin;
    end
```

DEMO

http://thevirtualheart.org/

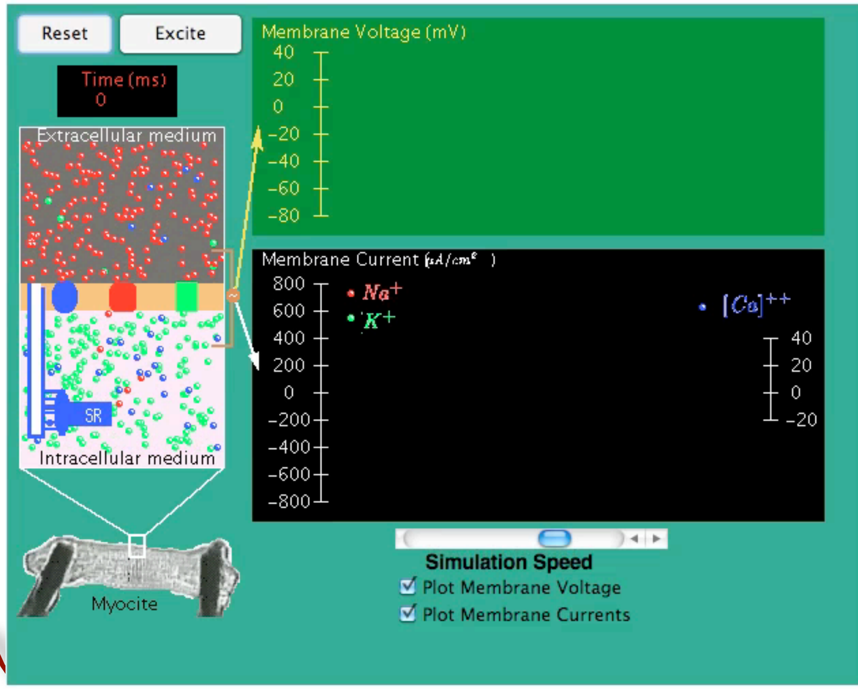
Virtual Heart Examples

Nerve Action Potential



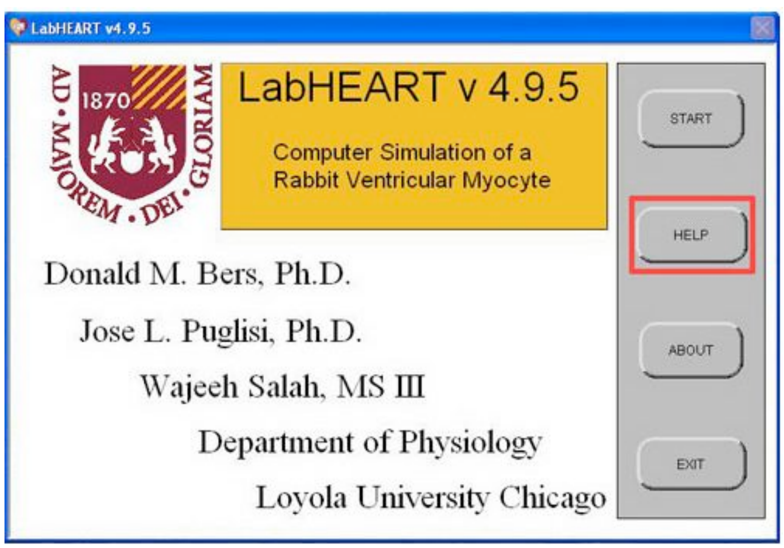
Virtual Heart Examples

Cardiac Action Potential



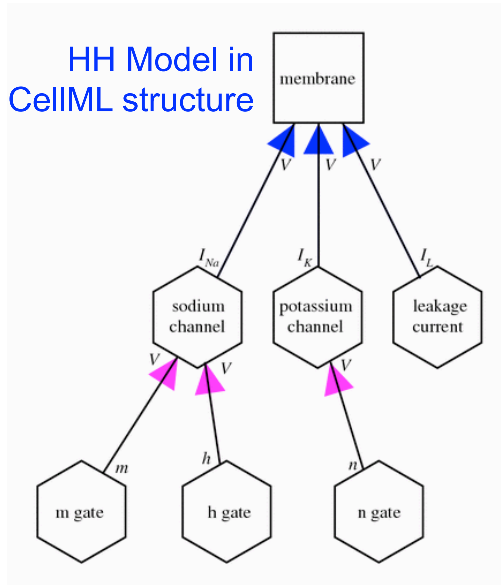
LabHeart

http://www.labheart.org



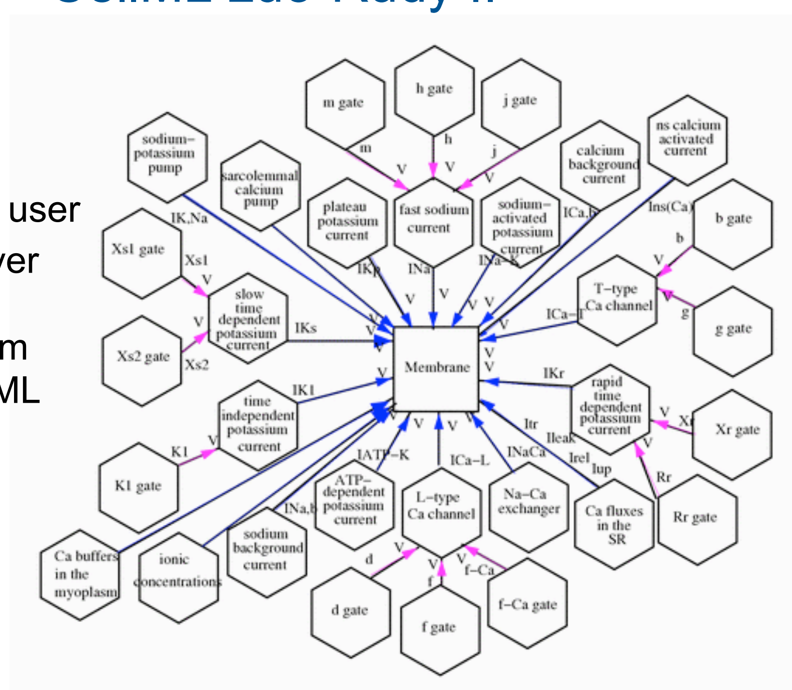
CellML

- www.cellml.org
- Standardized interface to cell simulations
- Includes geometry, electrophysiology and signal transduction
- Based on markup language (ML) standards
- Part of larger set of ML's for simulation



CellML Luo-Rudy II

- Additional complexity hidden from user
- Software layer provides interface from user to CellML functions



CellML + JSim

- Tcl script based interface to CellML functions