The NEURON Simulator

www.neuron.yale.edu

Supported by NINDS

Where does it fit in the overall modeling process?

Using:

- **Building**
- **Running**
- **Analysing**

How is the program organized.

Where does it fit in the overall modeling process?

Using:

- **Building** **What kind of models?**
- **Running**
- **Analysing**

How is the program organized.

Where does it fit in the overall modeling process?

Using:

Building	**What kind of models?**
Running	**Numerical methods**
Analysing	

How is the program organized.

Where does it fit in the overall modeling process?

Using:

Building	**What kind of models?**
Running	**Numerical methods**
Analysing	

How is the program organized.

Help!

Physical System

Model

Representation in NEURON

Create Representation

Investigate/Explore/Control/Use Representation

Physical System

Model

Simulation

Squid axon

Hodgkin−Huxley cable equations

$$\frac{D}{4R_a} \cdot \frac{\partial^2 V}{\partial x^2} = C_m \frac{\partial V}{\partial t} + \overline{g}_{na} m^3 h \cdot (V - E_{na}) + \overline{g}_k n^4 \cdot (V - E_k) + g_l \cdot (V - E_l)$$

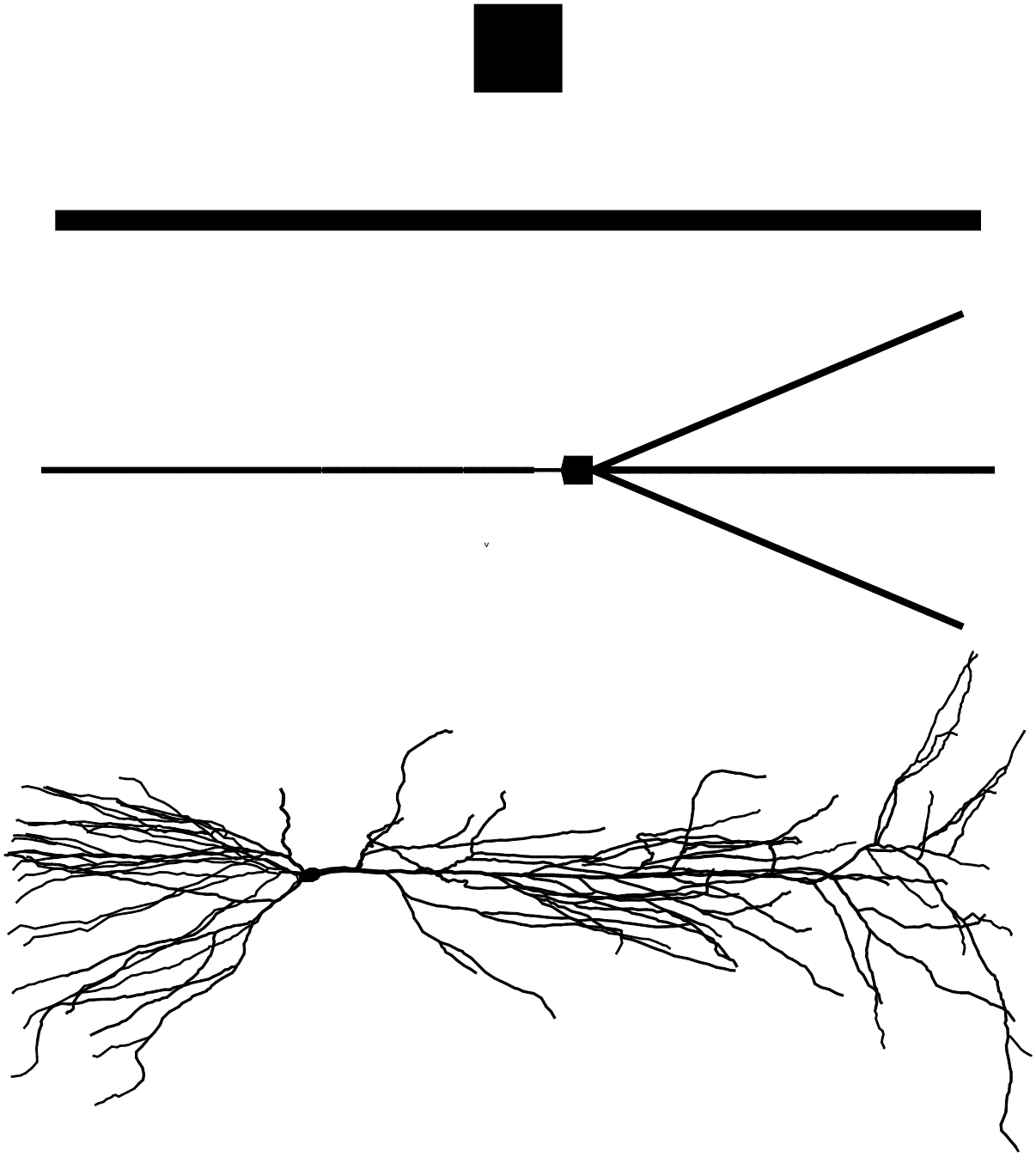
$$\frac{dm}{dt} = -\alpha_m m + \beta_m \cdot (1 - m) \quad \alpha_m = \frac{.1(V+40)}{1-e^{-.1(V+40)}} \quad \beta_m = 4e^{-(V+65)/18)}$$

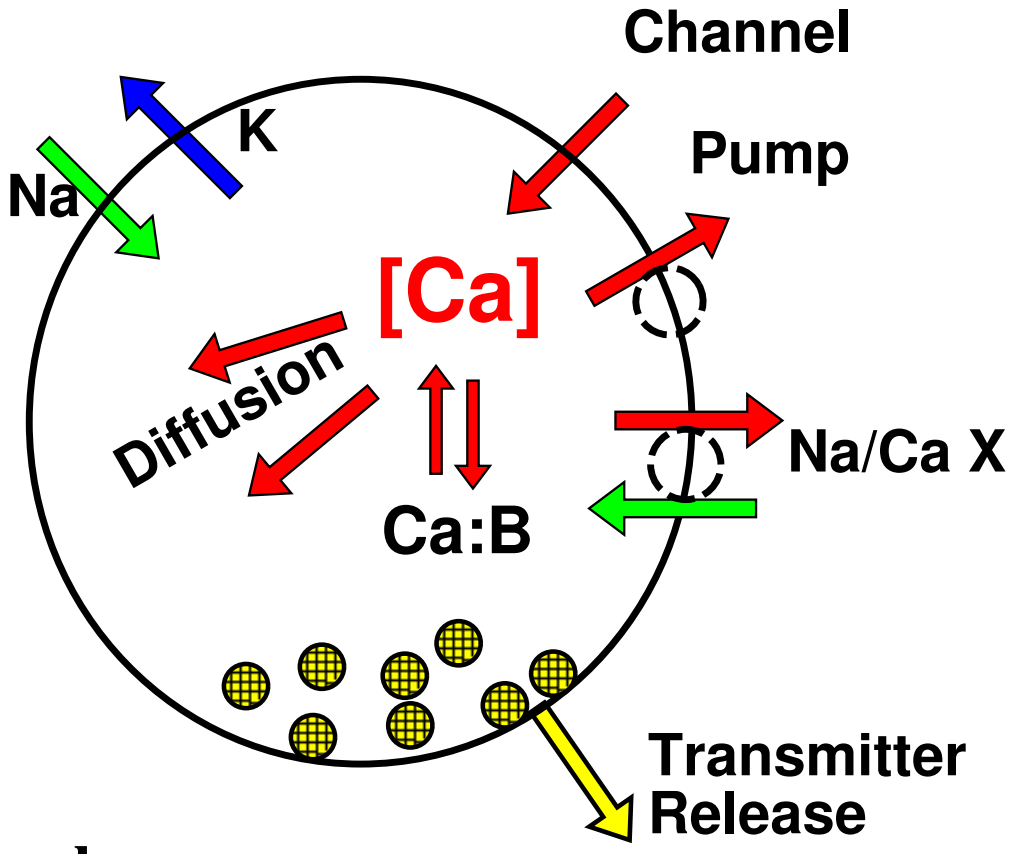
$$\frac{dh}{dt} = -\alpha_h h + \beta_h \cdot (1 - h) \quad \alpha_h = .07e^{-.05(V+65)} \quad \beta_h = \frac{1}{1+e^{-.1(V+35)}}$$

$$\frac{dn}{dt} = -\alpha_n n + \beta_n \cdot (1 - n) \quad \alpha_n = \frac{.01(V+55)}{1-e^{-.1(V+55)}} \quad \beta_n = .125e^{-(V+65)/80}$$

NEURON representation

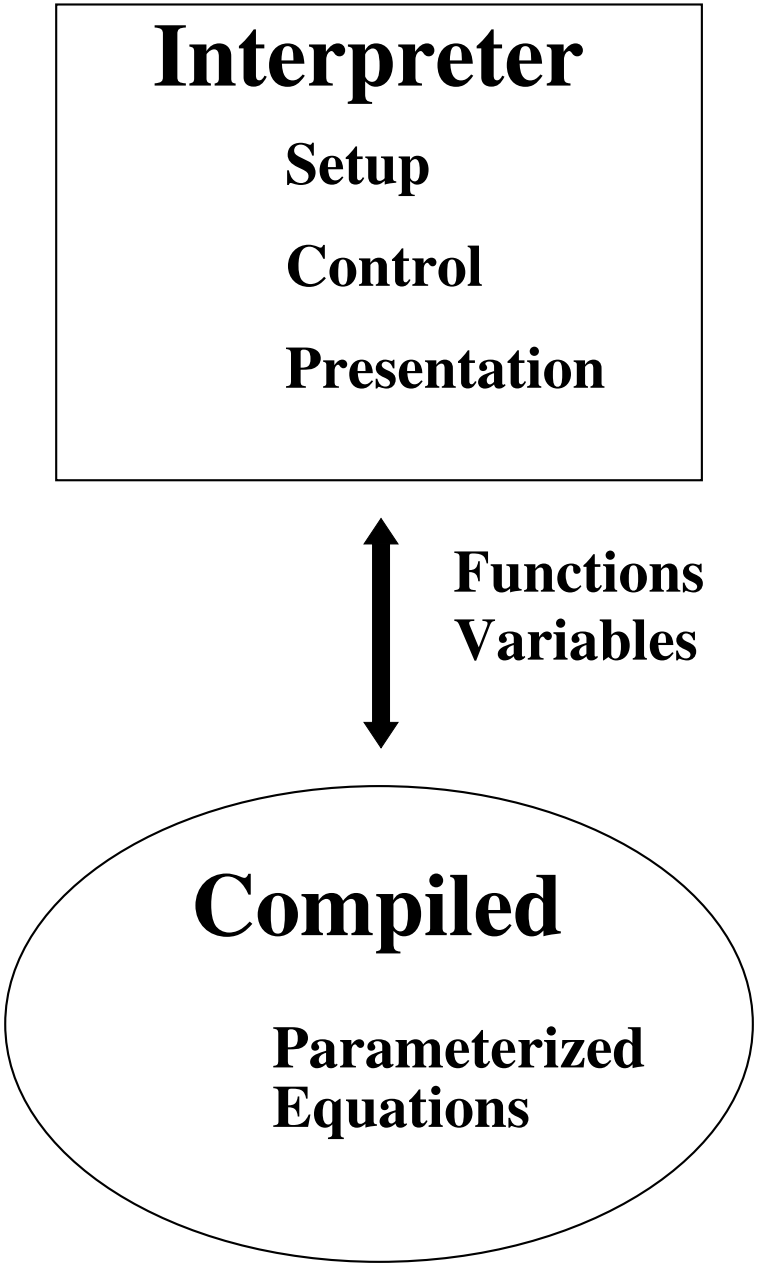
```
create axon
axon {
   nseg = 75
   diam = 100
   L = 20000
   insert hh
}
```

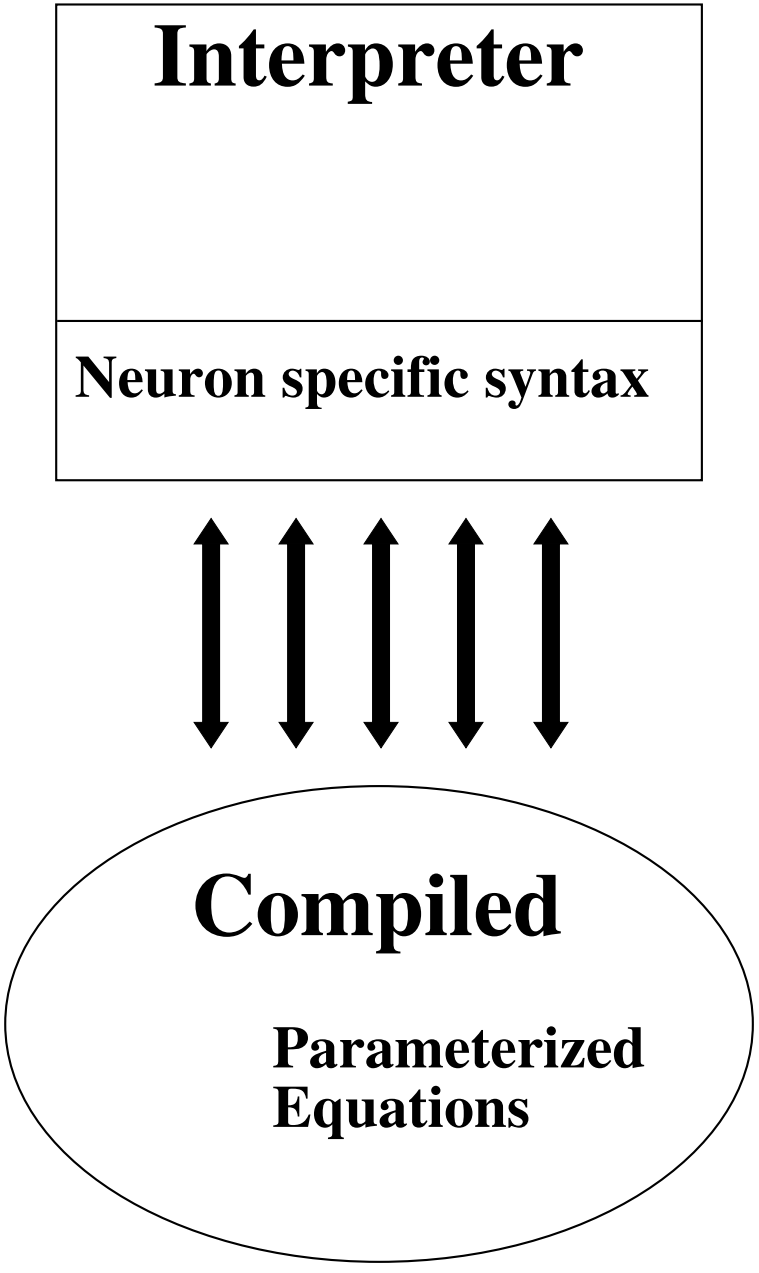


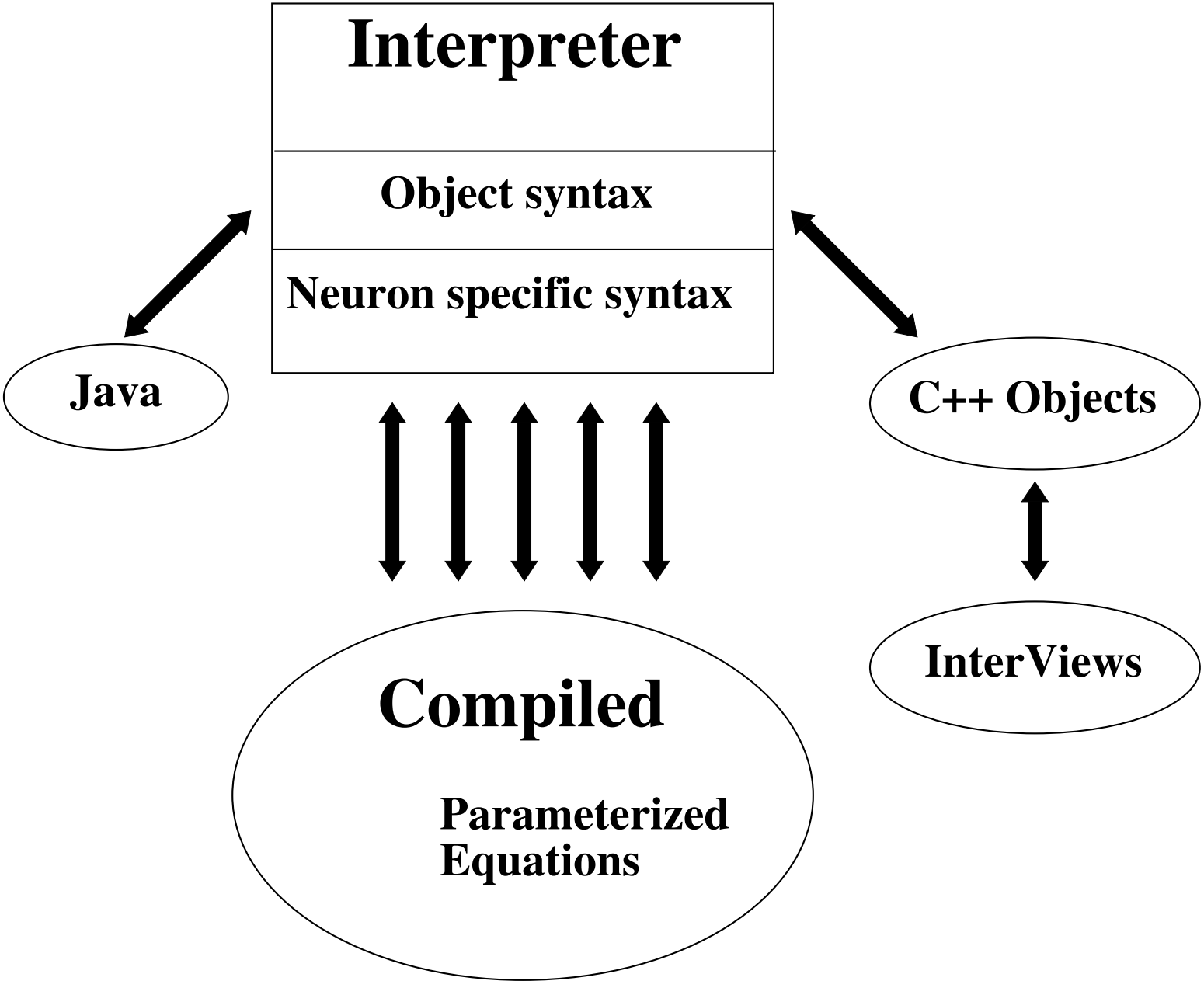


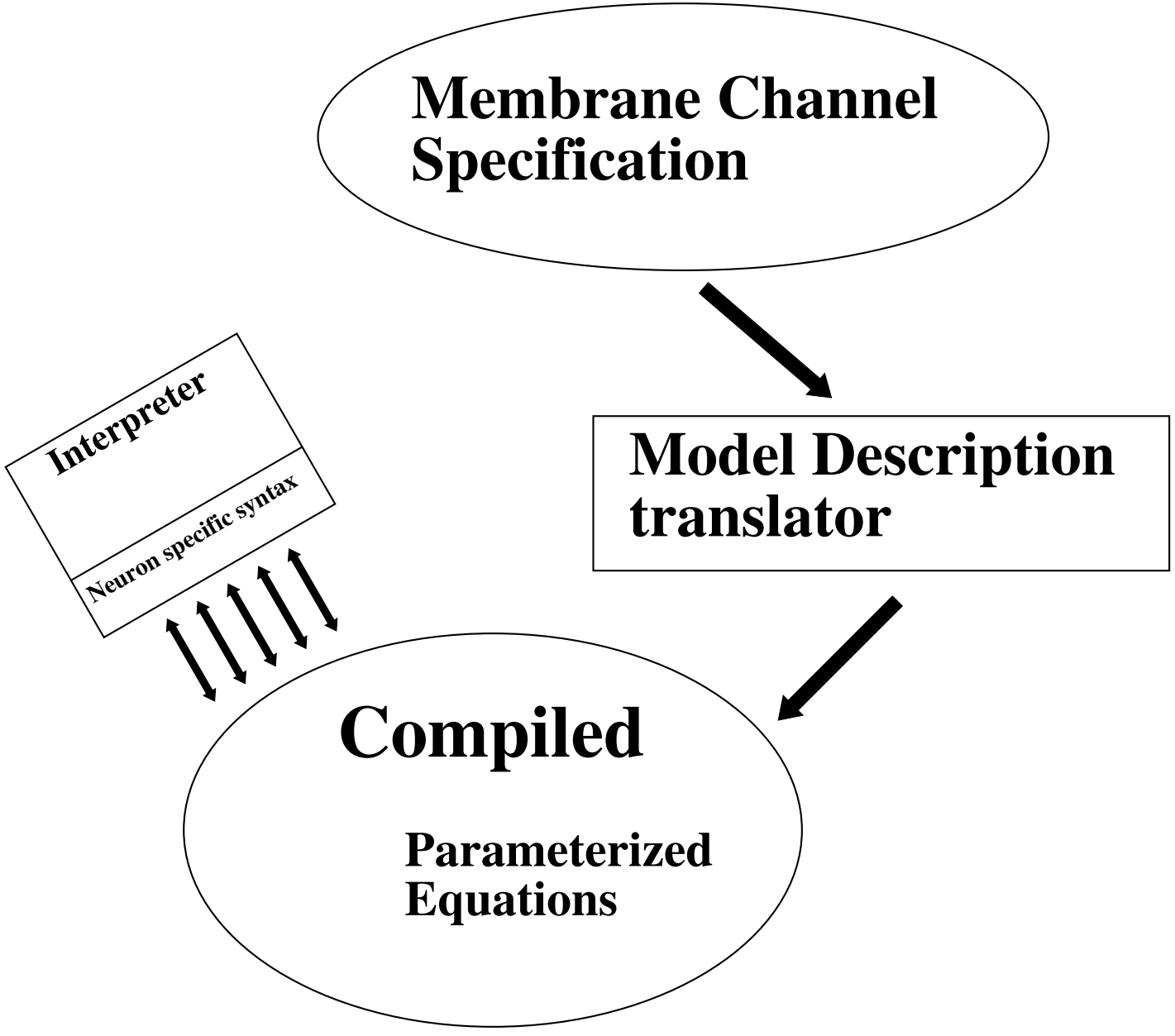
Single Channels
Extracellular fields
Linear circuits
Synapses
Networks

Discrete Event Simulation
Artificial Spiking Cells

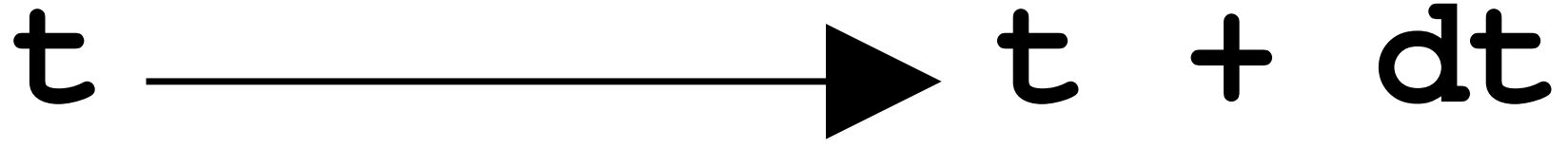

Interpreter
Setup
Control
Presentation
Functions
Variables
Compiled
Parameterized
Equations


Interpreter
Neuron specific syntax
Compiled
Parameterized
Equations


Interpreter
Object syntax
Neuron specific syntax
Java
C++ Objects
InterViews
Compiled
Parameterized
Equations


Membrane Channel Specification
Model Description translator
Interpreter
Neuron specific syntax
Compiled
Parameterized Equations

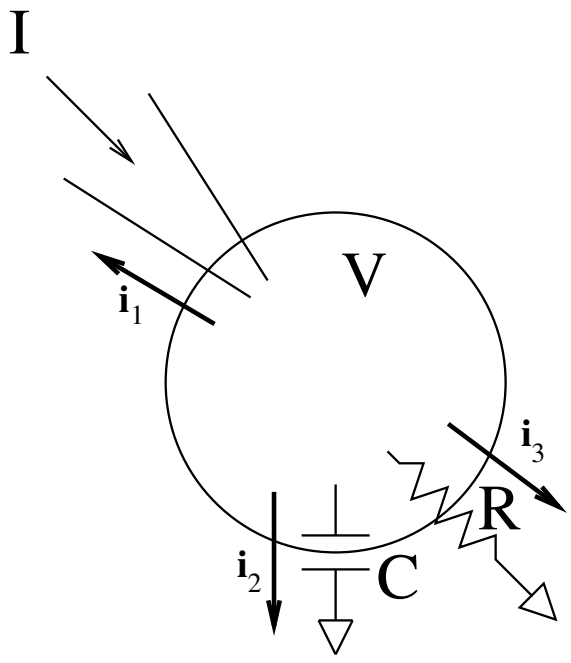
```
proc run() {
  finitialize(-65)
  while (t < tstop) {
    fadvance()
  }
}
```

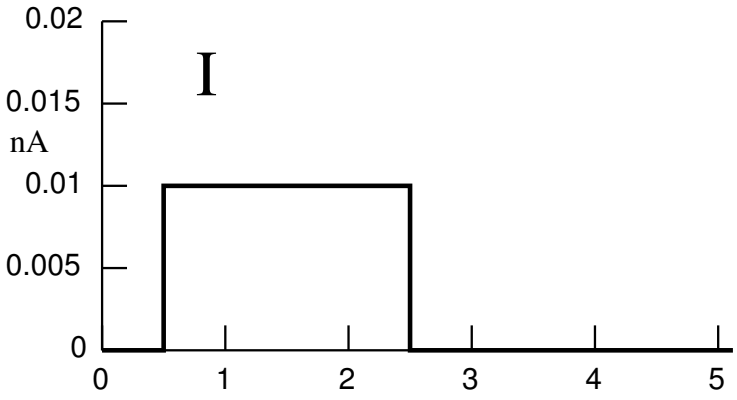


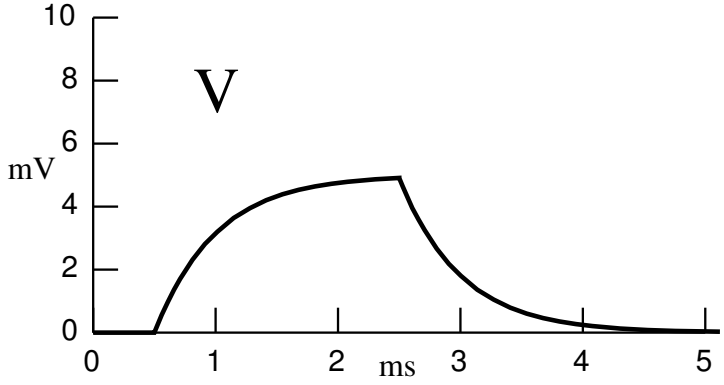
Compartmental Modeling

Not much mathematics required.

Good judgment essential!







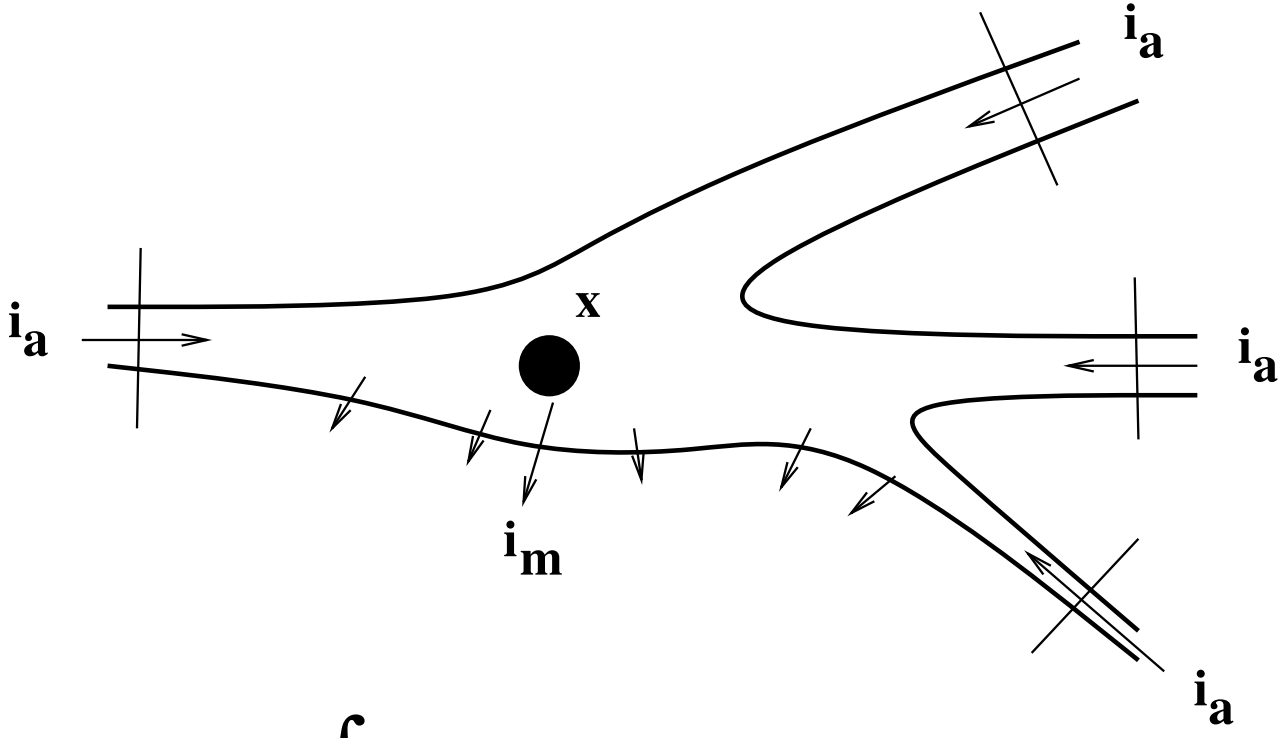
$$i_1 + i_2 + i_3 = 0$$

$$i_1 = -I$$

$$i_2 = C\ dV/\ dt$$

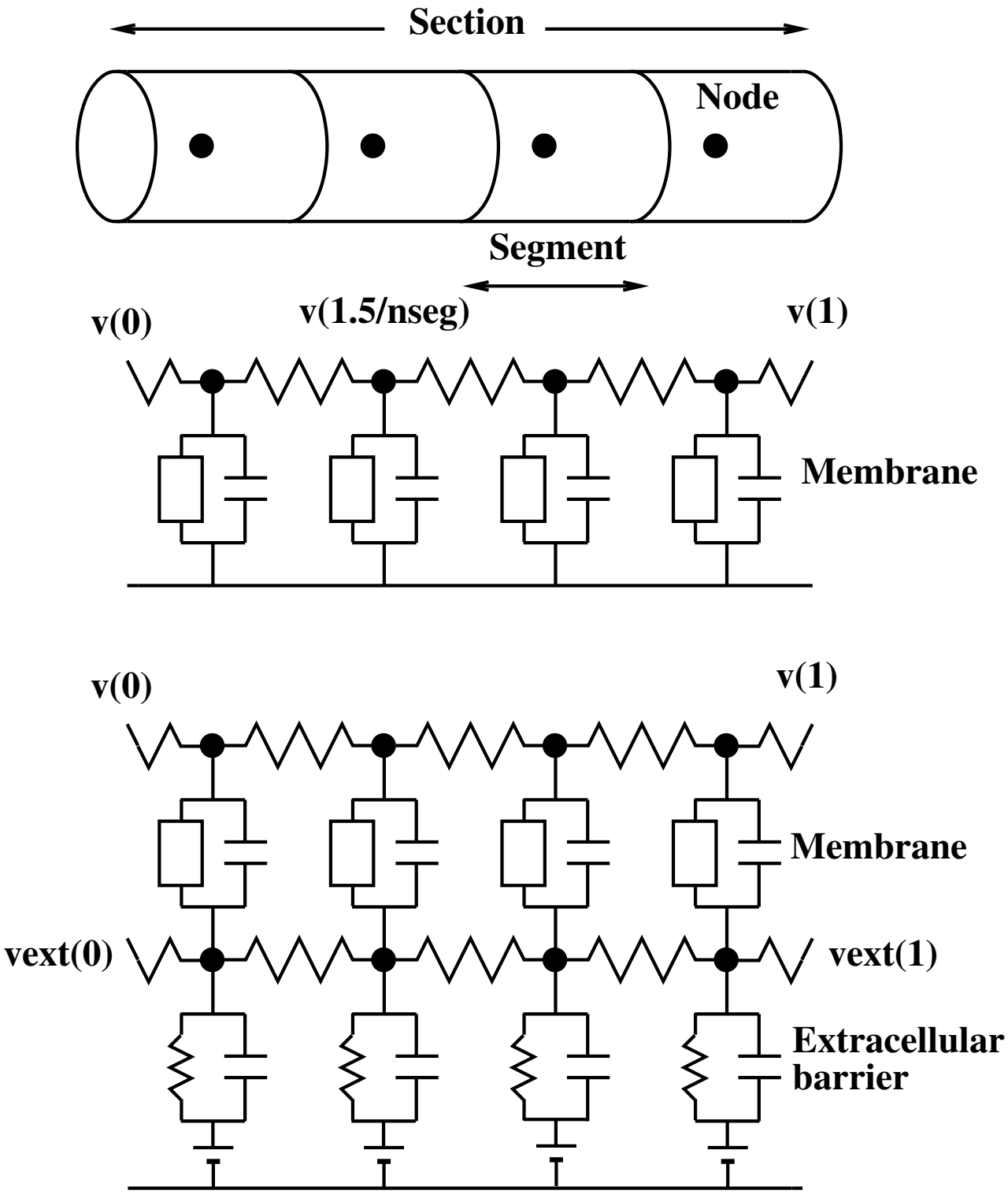
$$i_3 = V/\ R$$

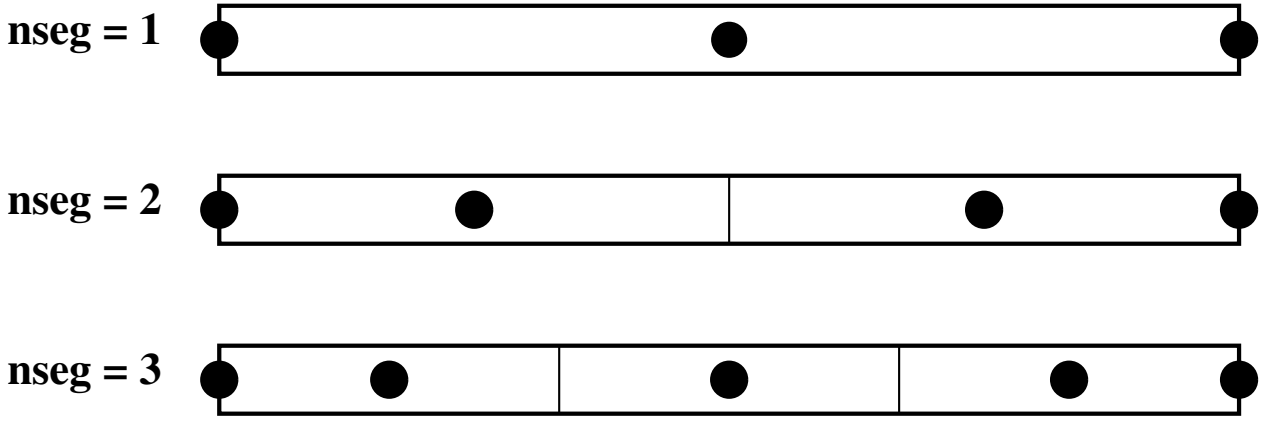
$C\ dV/dt + g\ V = I$



$$\int \mathbf{i}_m = \sum \mathbf{i}_a$$

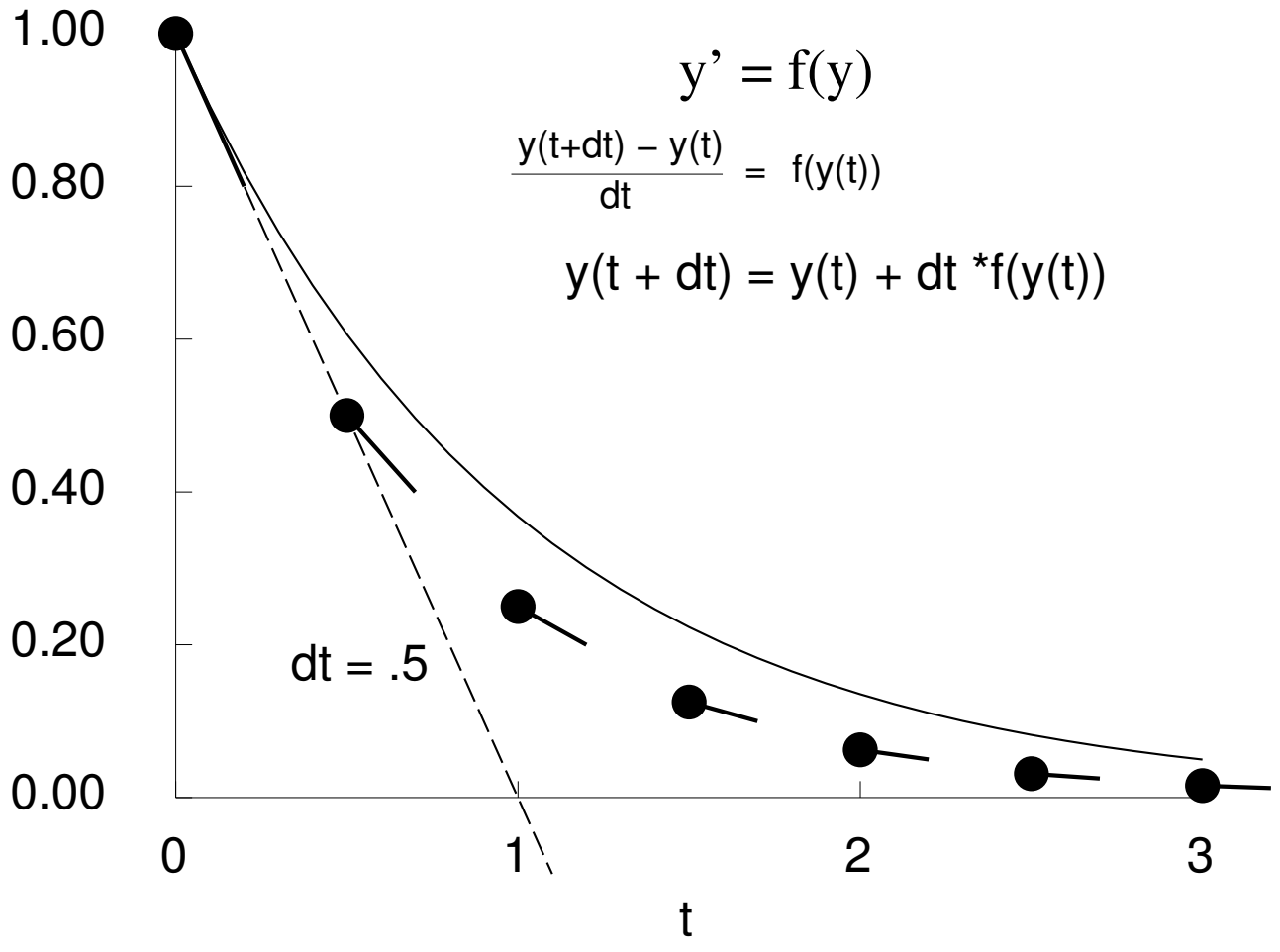
$$\mathbf{c}_j \frac{\mathbf{d}\,\mathbf{v}_j}{\mathbf{dt}} + \mathbf{i}_j = \sum_{\mathbf{k}} \frac{\mathbf{v}_k - \mathbf{v}_j}{\mathbf{r}_{jk}}$$

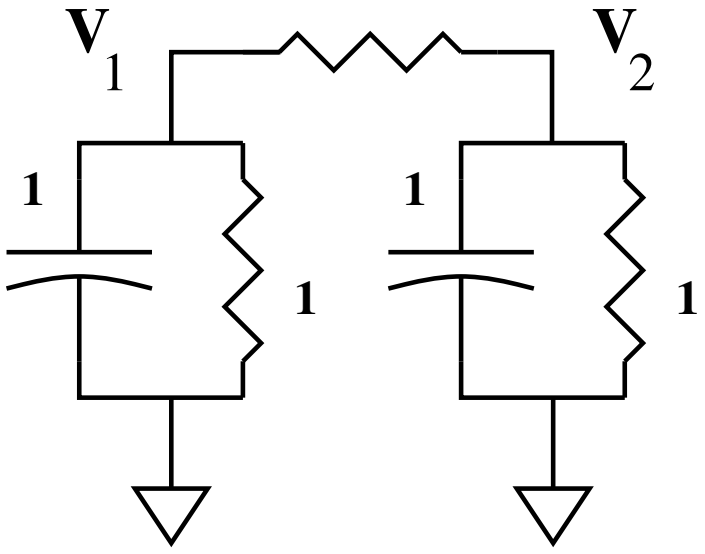

Section
Node
Segment
v(0)
v(1.5/nseg)
v(1)
Membrane
v(0)
v(1)
Membrane
vext(0)
vext(1)
Extracellular
barrier

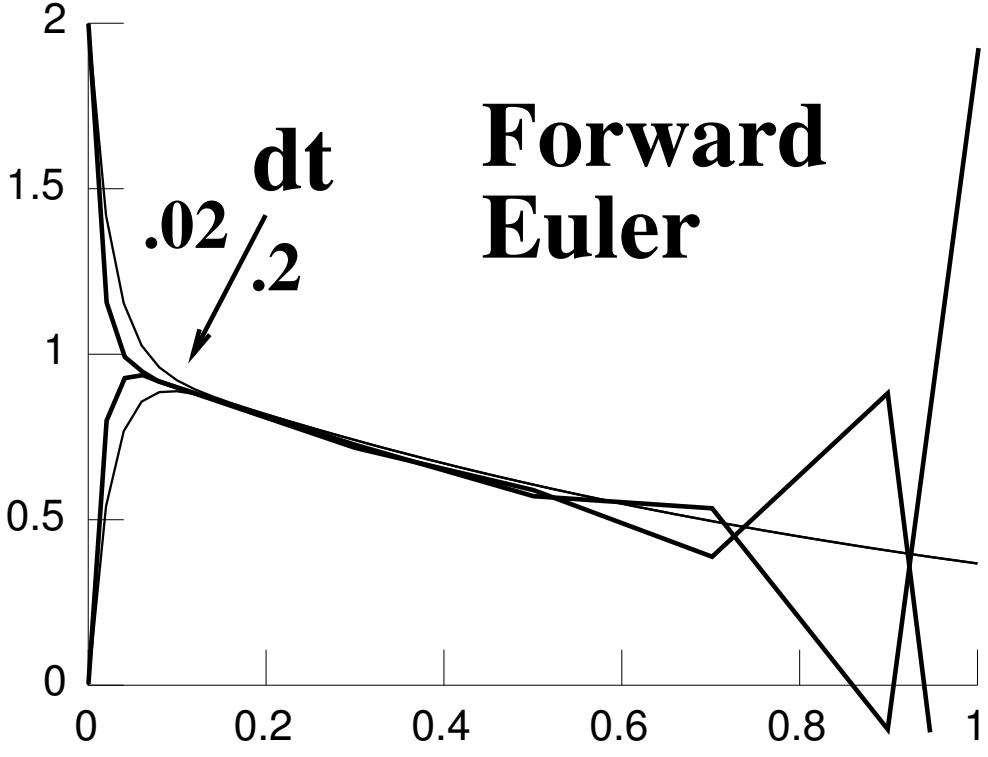



```
forall nseg *= 3
```

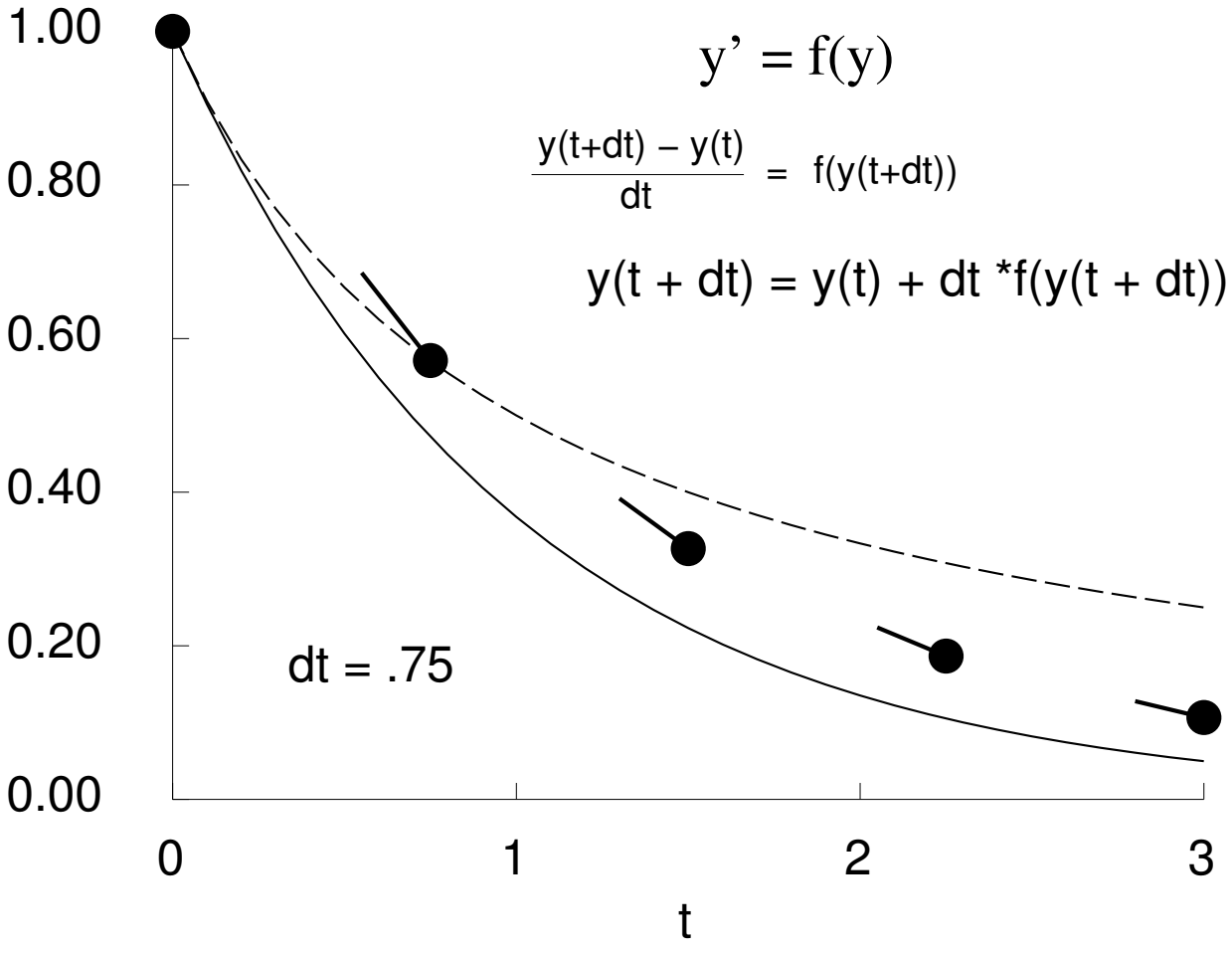
Forward Euler



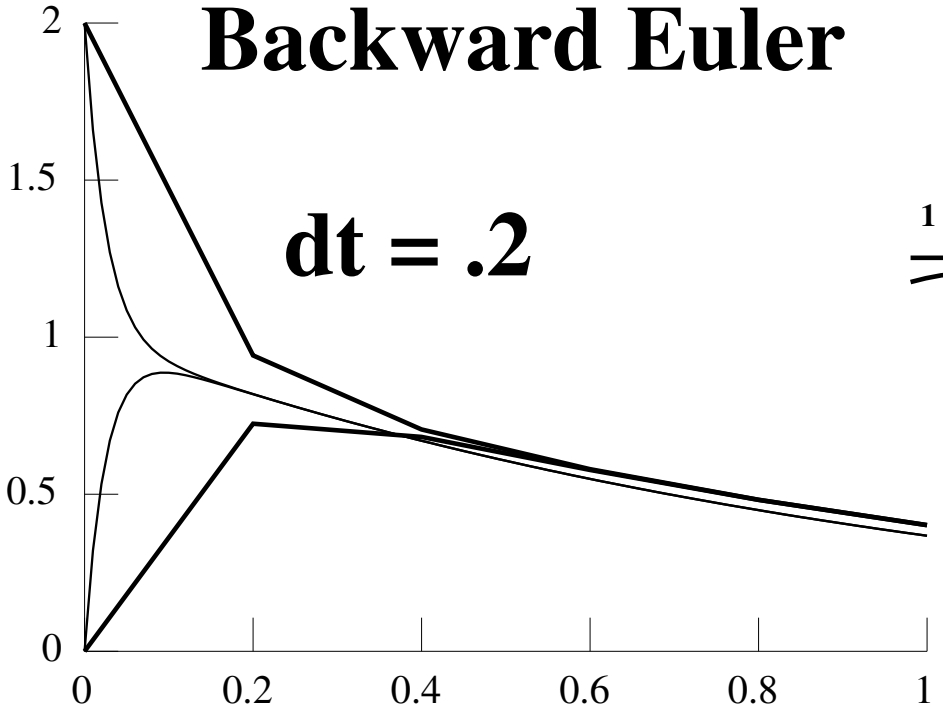

V1
V2
1
1
1
1

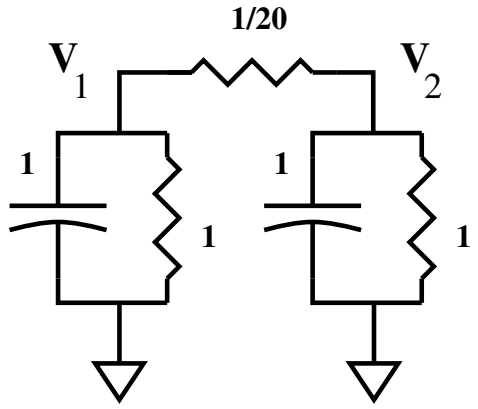

dt
.02
.2
Forward
Euler
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

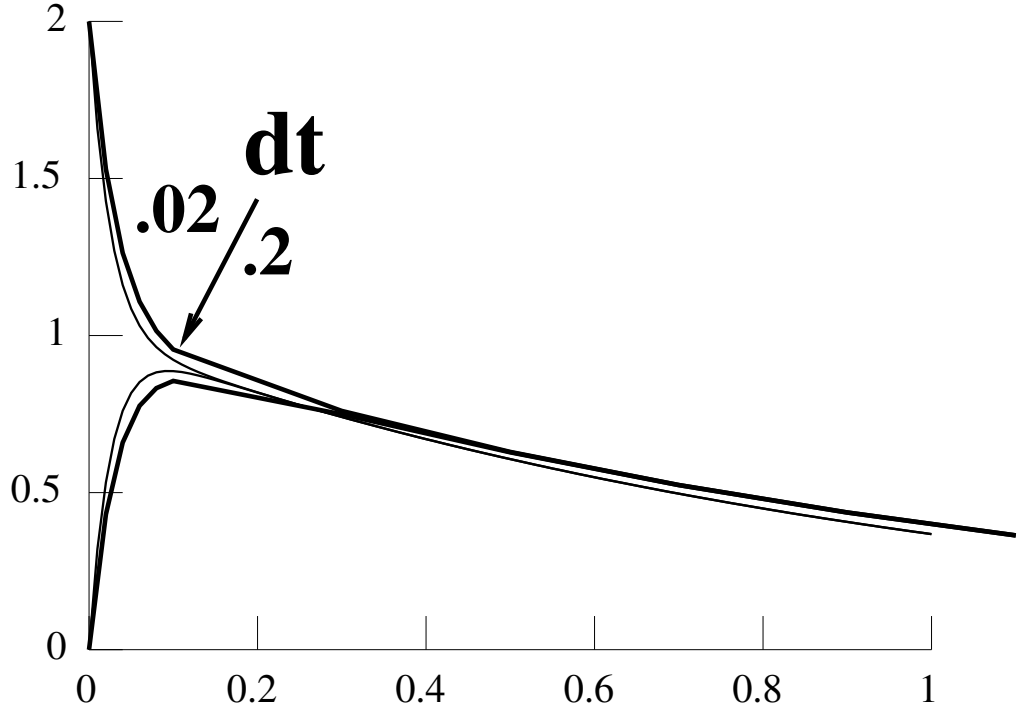
Backward Euler



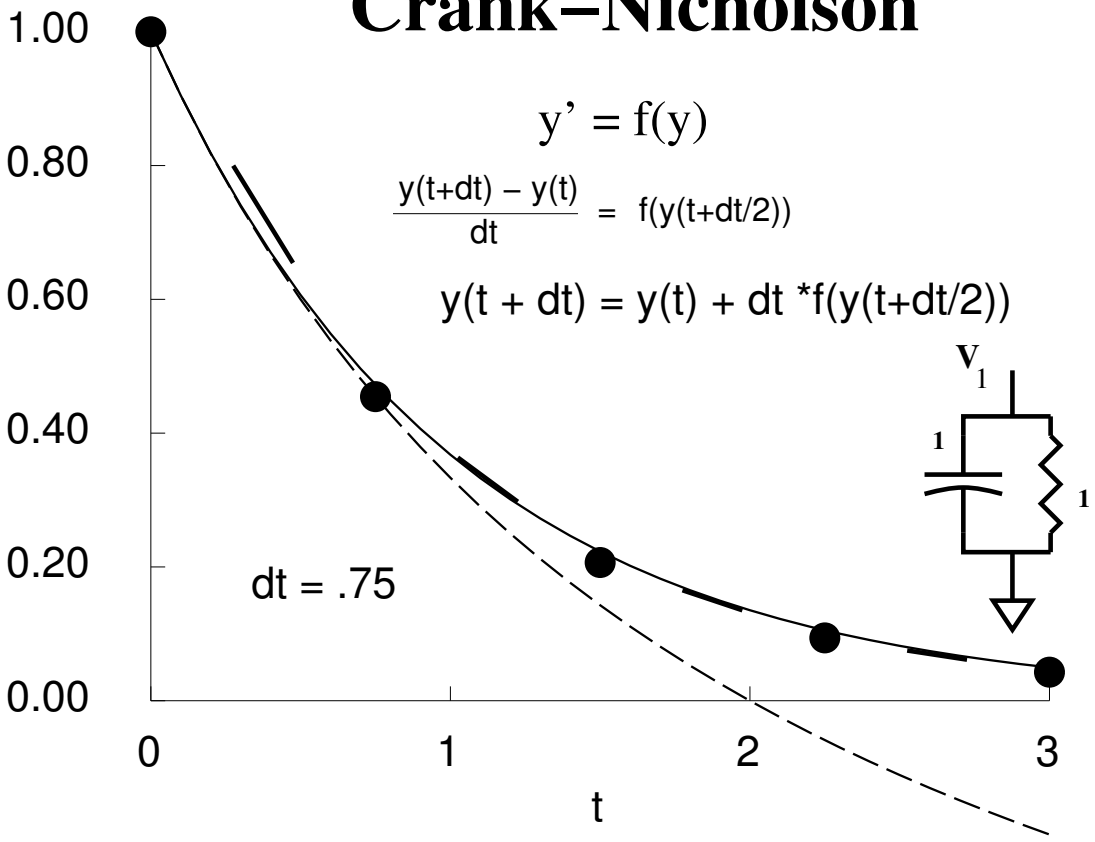
Backward Euler







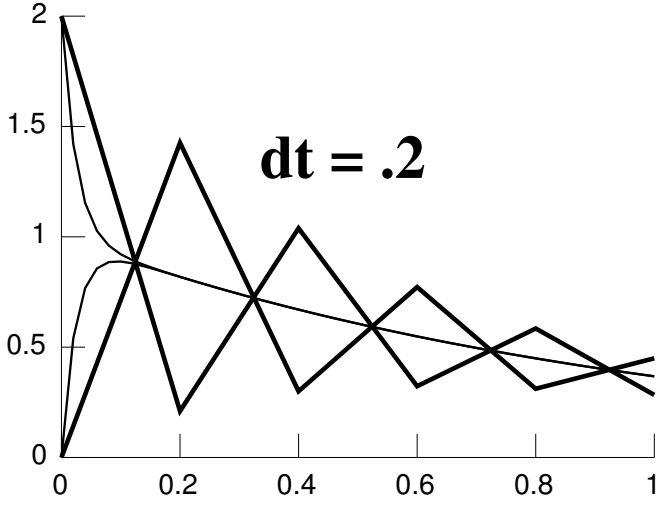
Crank–Nicholson

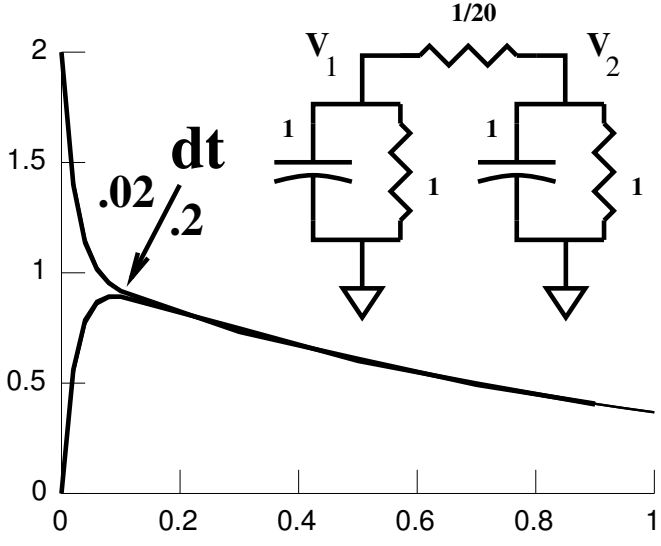


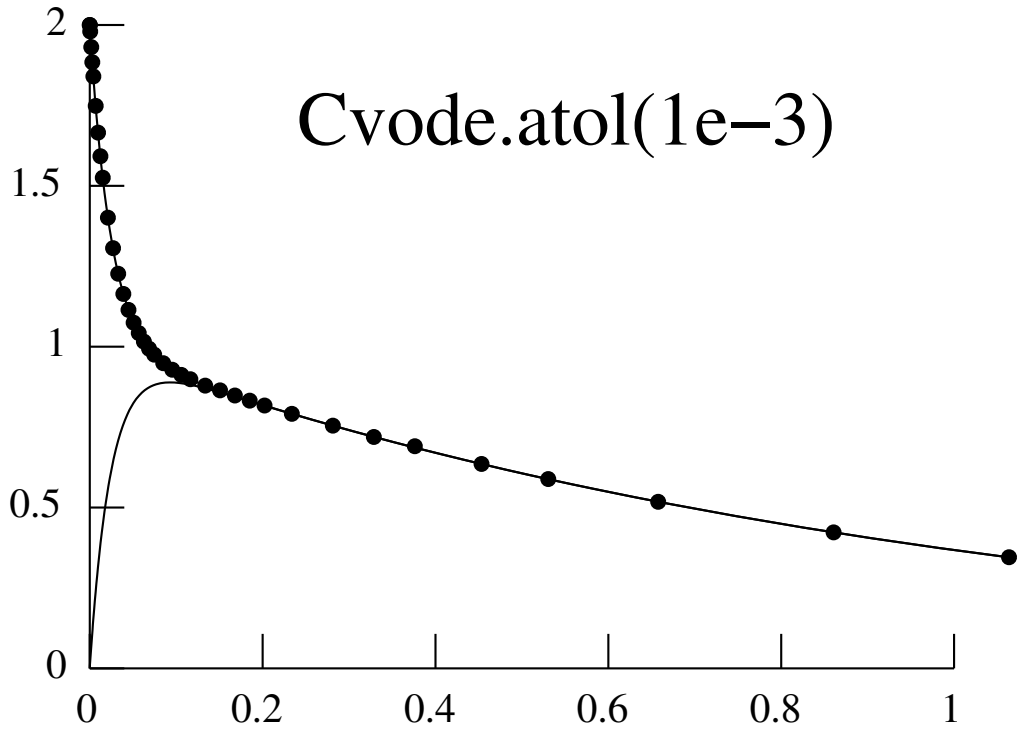
$$y' = f(y)$$

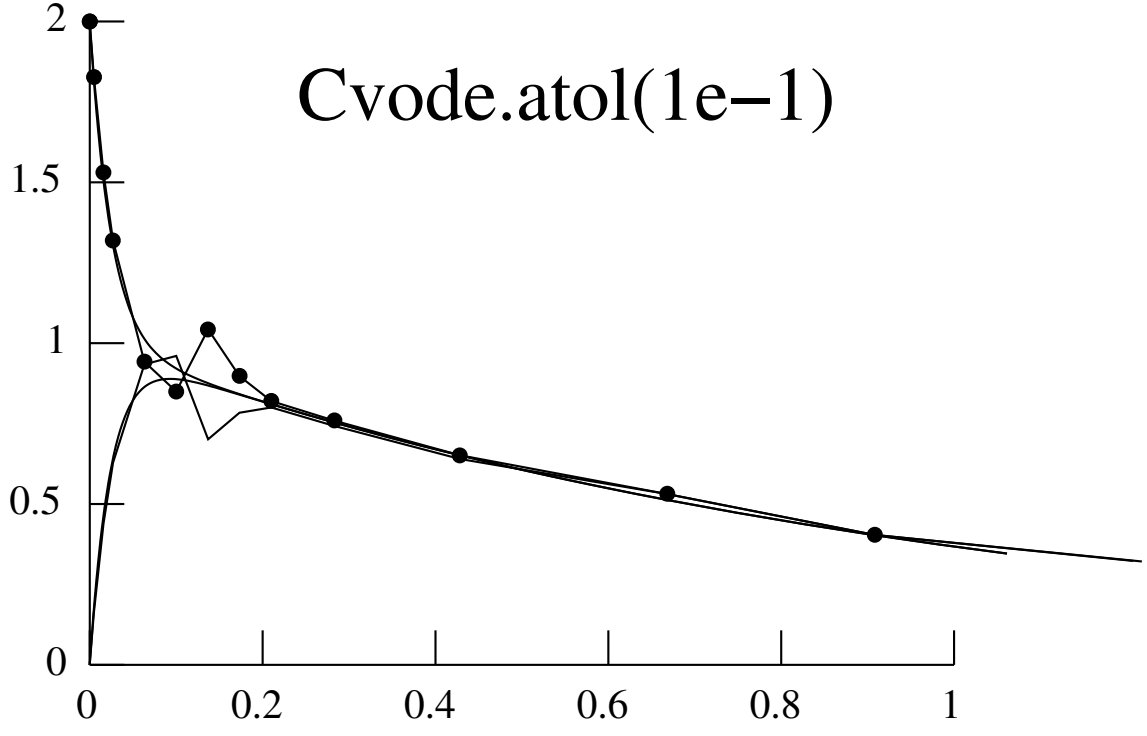
$$\frac{y(t+dt) - y(t)}{dt} = f(y(t+dt/2))$$

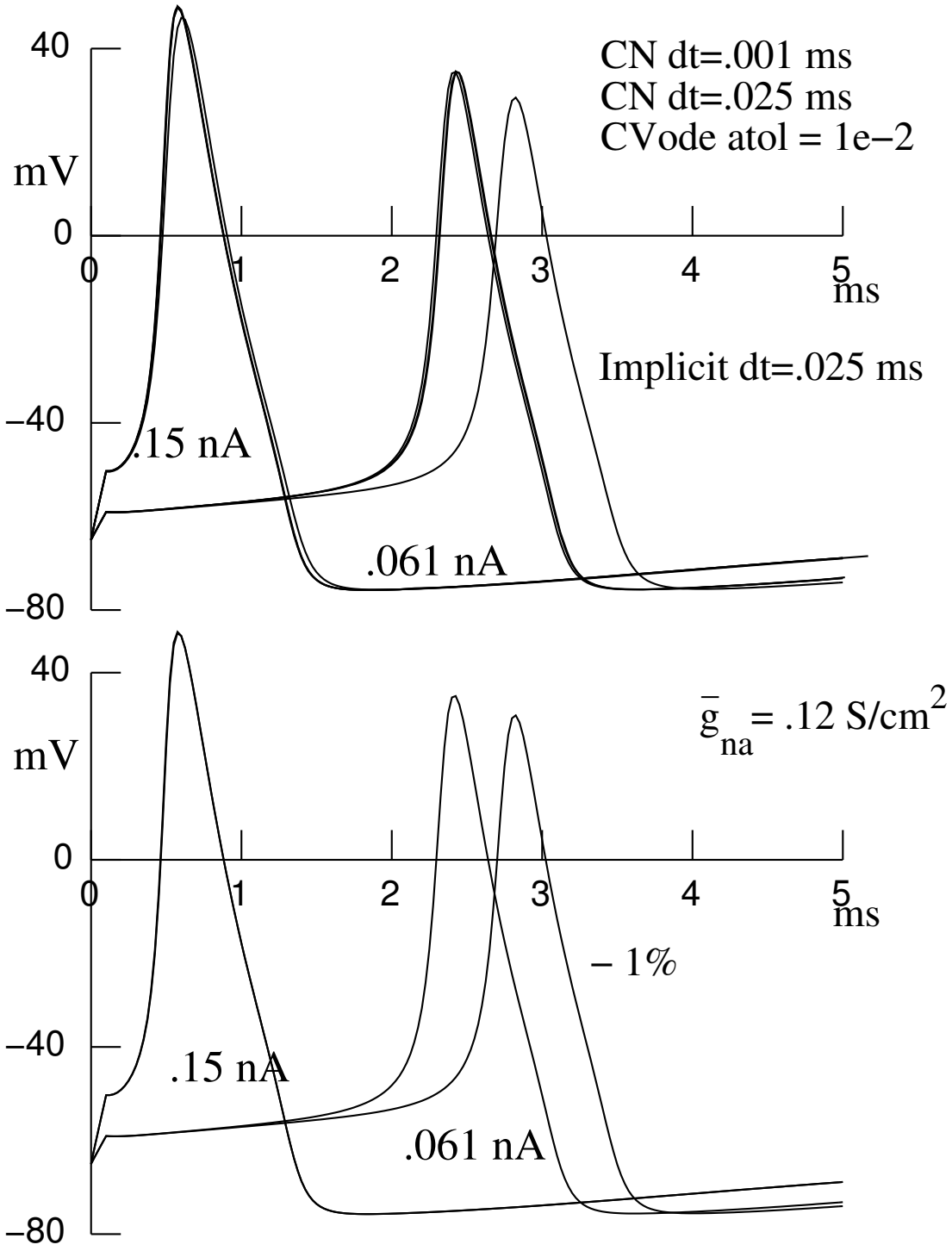
$$y(t + dt) = y(t) + dt * f(y(t+dt/2))$$

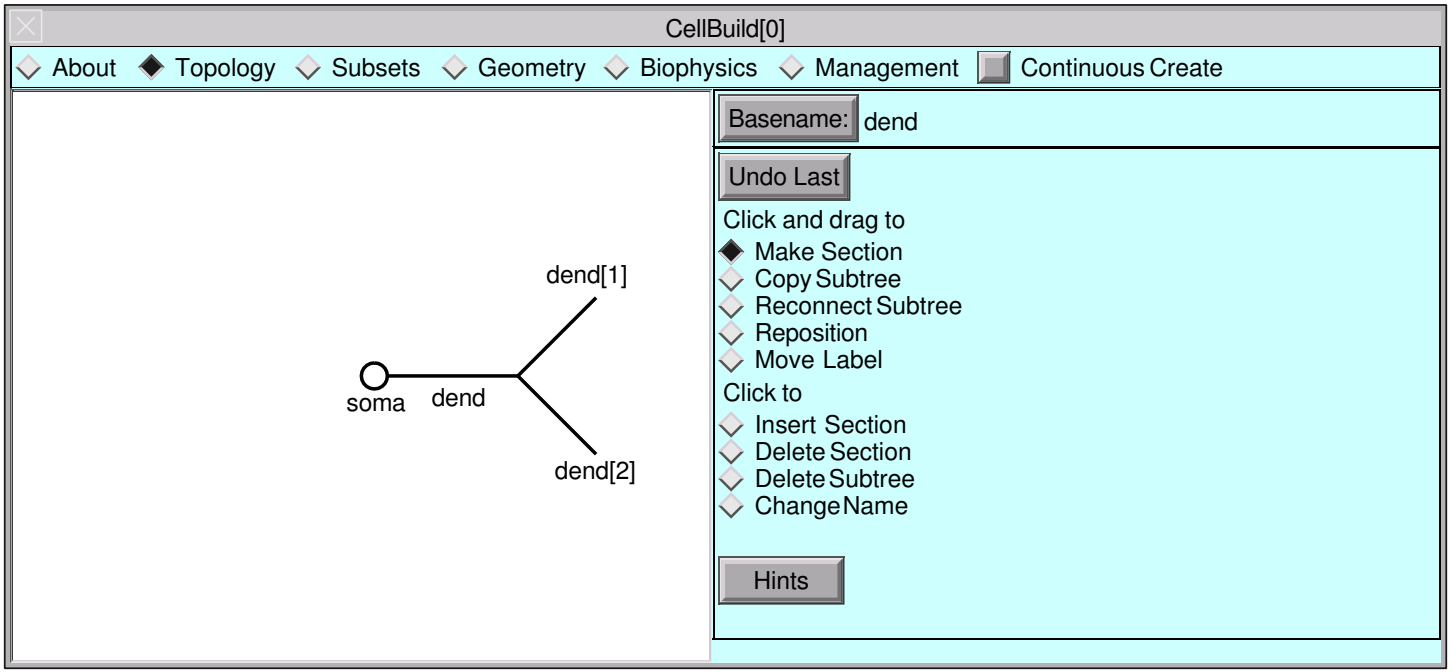


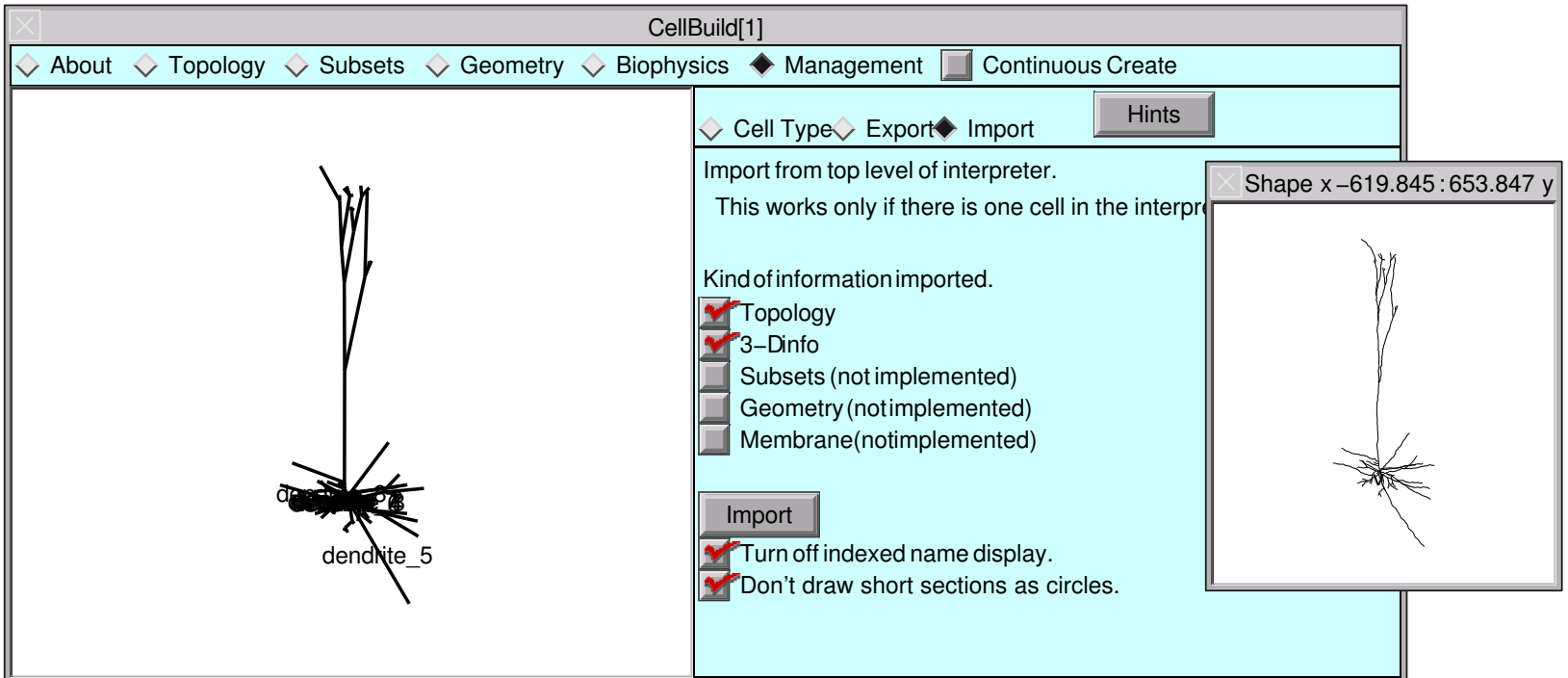



Cvode.atol(1e−3)
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1


Cvode.atol(1e−1)
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1


40
mV
0
−40
−80
0
1
2
3
4
5
ms
CN dt=.001 ms
CN dt=.025 ms
CVode atol = 1e−2
Implicit dt=.025 ms
.15 nA
.061 nA
40
mV
0
−40
−80
0
1
2
3
4
5
ms
$\bar{g}_{na}$ = .12 S/cm^2
− 1%
.15 nA
.061 nA


CellBuild[0]
About
Topology
Subsets
Geometry
Biophysics
Management
Continuous Create
Basename:
dend
Undo Last
Click and drag to
Make Section
Copy Subtree
Reconnect Subtree
Reposition
Move Label
Click to
Insert Section
Delete Section
Delete Subtree
Change Name
Hints
dend[1]
soma
dend
dend[2]


CellBuild[1]
About
Topology
Subsets
Geometry
Biophysics
Management
Continuous Create
Cell Type
Export
Import
Hints
Import from top level of interpreter.
This works only if there is one cell in the interpr
Kind of information imported.
Topology
3–Dinfo
Subsets (not implemented)
Geometry (not implemented)
Membrane (not implemented)
Import
Turn off indexed name display.
Don't draw short sections as circles.
dendrite_5
Shape x –619.845 : 653.847 y

PointProcessManager

Close Hide

SelectPointProcess

none
IClamp
AlphaSynapse
ExpSyn
Exp2Syn
SEClamp
VClamp
OClamp
APCount
NetStim
IntFire1
IntFire2
IntFire4
PointProcessMark
IntervalFire

PointProcessManager

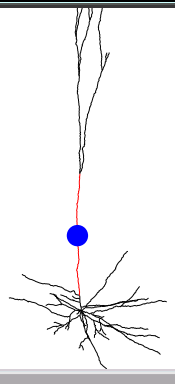
Close Hide

SelectPointProcess

Show

IClamp[2]

at:dendrite_1[8](0.5)



PointProcessManager

Close Hide

SelectPointProcess

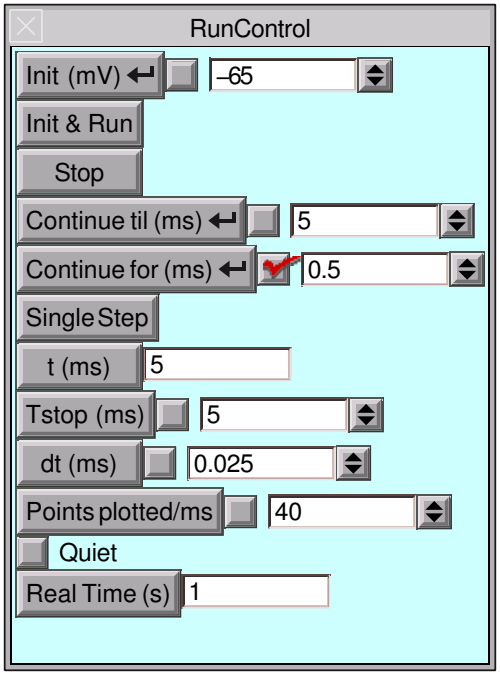
Show

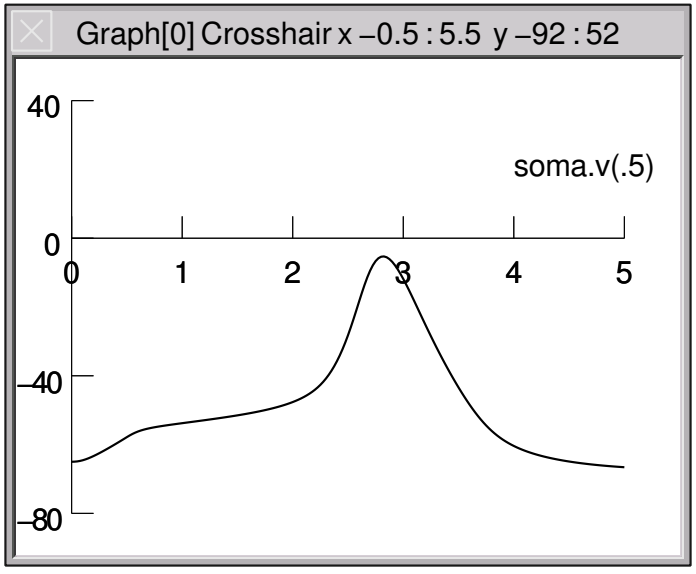
IClamp[1]

at:soma(0.5)

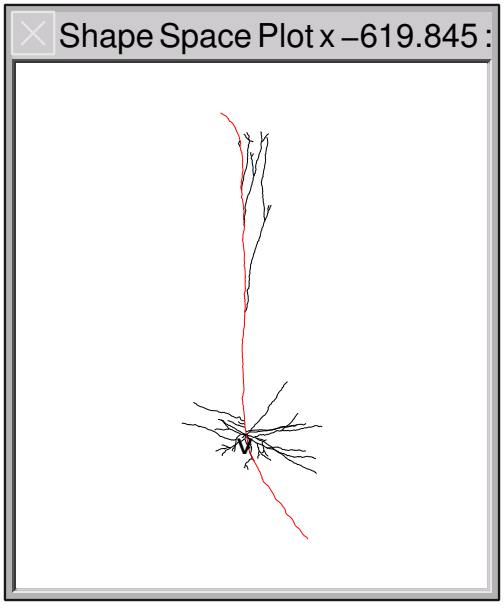
IClamp[1]

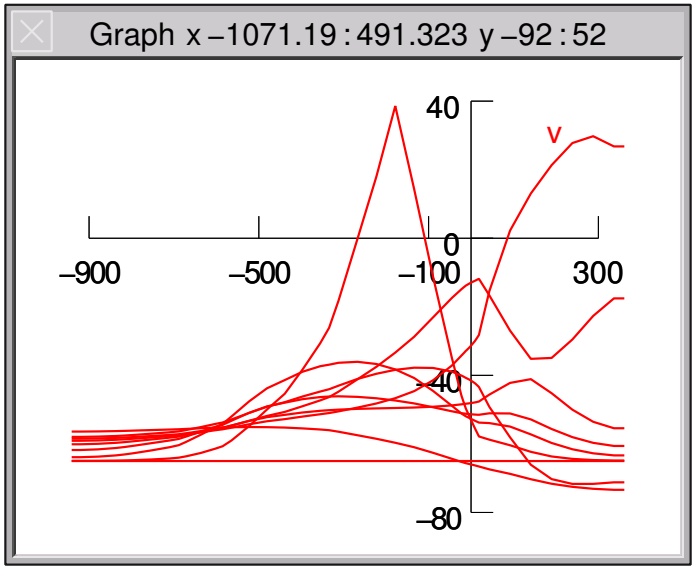
del (ms)	1
dur (ms)	2
amp (nA)	25
i (nA)	0

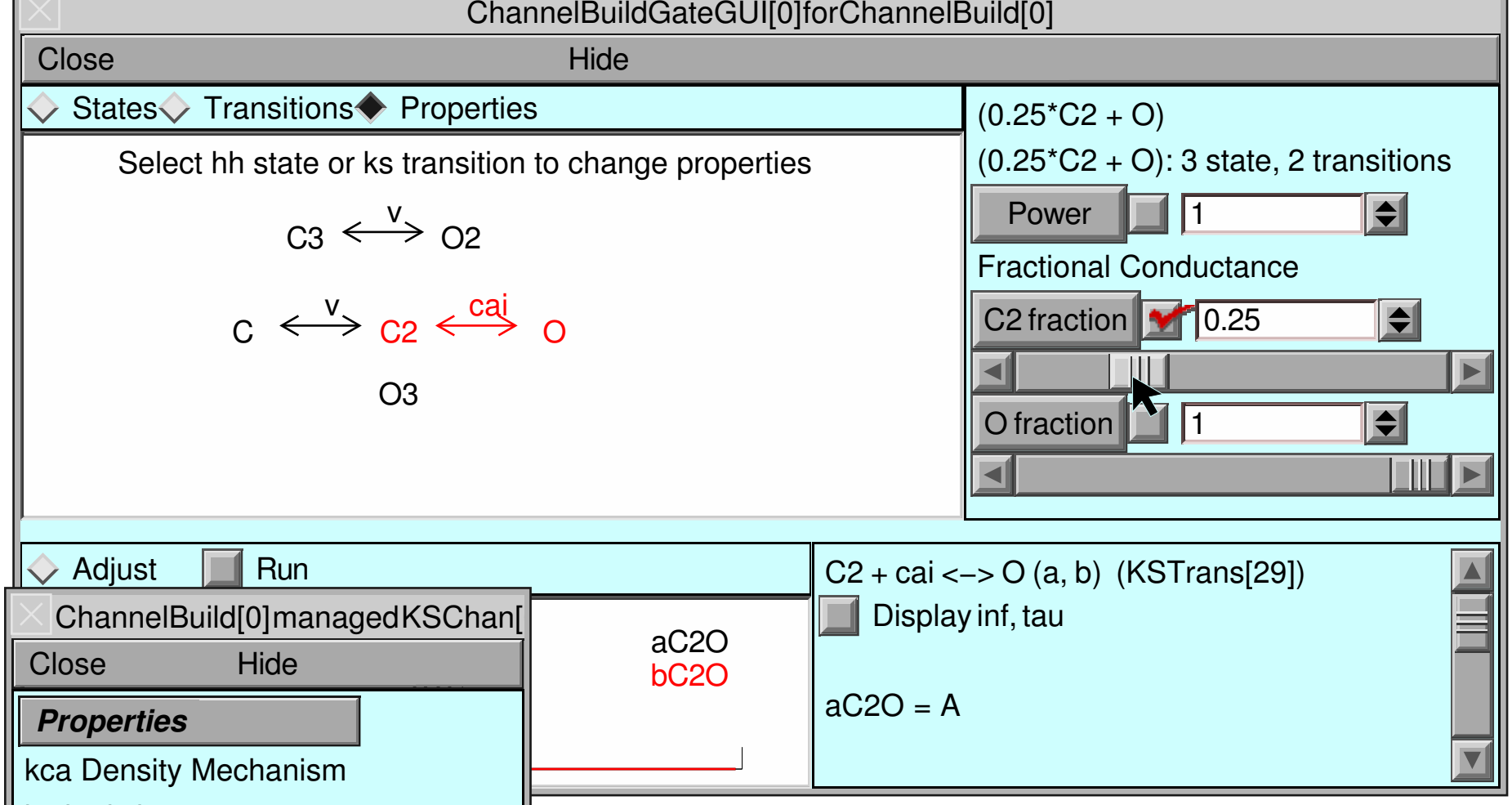

RunControl
Init (mV)
–65
Init & Run
Stop
Continue til (ms)
5
Continue for (ms)
0.5
Single Step
t (ms)
5
Tstop (ms)
5
dt (ms)
0.025
Points plotted/ms
40
Quiet
Real Time (s)
1


Graph[0] Crosshair x –0.5 : 5.5 y –92 : 52
40
0
–40
–80
soma.v(.5)
0
1
2
3
4
5

$$cy' = f(t, y)$$


Shape Space Plot x –619.845 :


Graph x –1071.19 : 491.323 y –92 : 52
40
0
–40
–80
v
–900
–500
–100
300


ChannelBuildGateGUI[0]forChannelBuild[0]
Close
Hide
States
Transitions
Properties
Select hh state or ks transition to change properties
C3
v
O2
C
v
C2
cai
O
O3
(0.25*C2 + O)
(0.25*C2 + O): 3 state, 2 transitions
Power
1
Fractional Conductance
C2 fraction
0.25
O fraction
1
Adjust
Run
aC2O
bC2O
C2 + cai <–> O (a, b) (KSTrans[29])
Display inf, tau
aC2O = A
ChannelBuild[0]managedKSChan[
Close
Hide
Properties
kca Density Mechanism
k ohmic ion current
ik (mA/cm2) = g_kca * (v – ek)
g = gmax * (0.25*C2 + O) * O2 * O3
Default gmax = 0 (S/cm2)
(0.25*C2 + O): 3 state, 2 transiti
O2: 2 state, 1 transitions
O3' = aO3*(1 – O3) – bO3*O3

ChannelBuild[3] managed KSChan[3]

Close Hide

Properties

nahh0 Point Process (Allow Single Channels)

na ohmic ion current

ina (mA/cm2) = nahh0.g * (v – ena)*(0.01/area)

g = gmax * m3h1

Default gmax = 0.12 (uS)

m3h1: 8 state, 10 transitions

SingleComp

Close Hide

soma

pas

hh

nahh

khh

leak

PointProcessManager

Close Hide

SelectPointProcess

Show

nahh0[0]

at:soma(0.5)

nahh0[0]

Nsingle 0

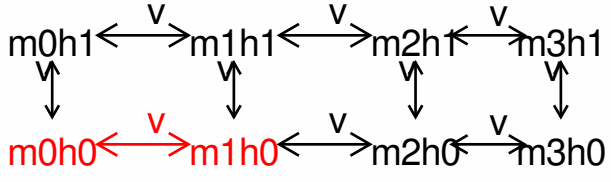
gmax (uS) 0.12

ChannelBuildGateGUI[0] forChannelBuild[3]

Close Hide

States Transitions Properties

Select hh state or ks transition to change properties



m3h1

m3h1: 8 state, 10 transitions

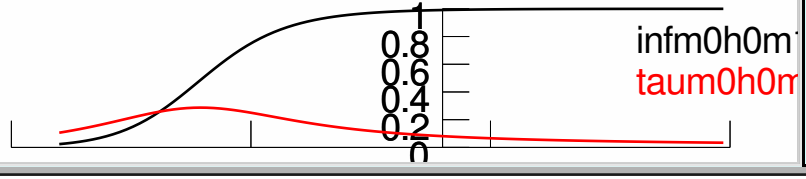
Power 1

Fractional Conductance

m0h0 fraction 0

m1h0 fraction 0

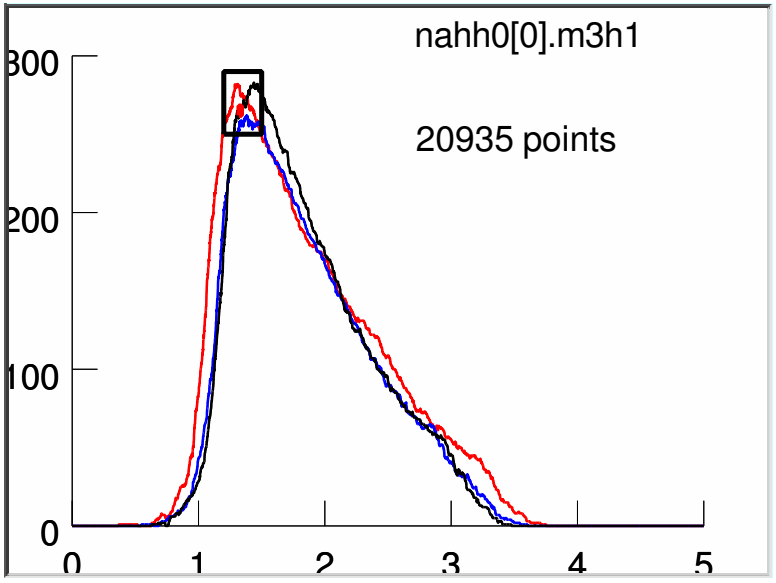
Adjust Run



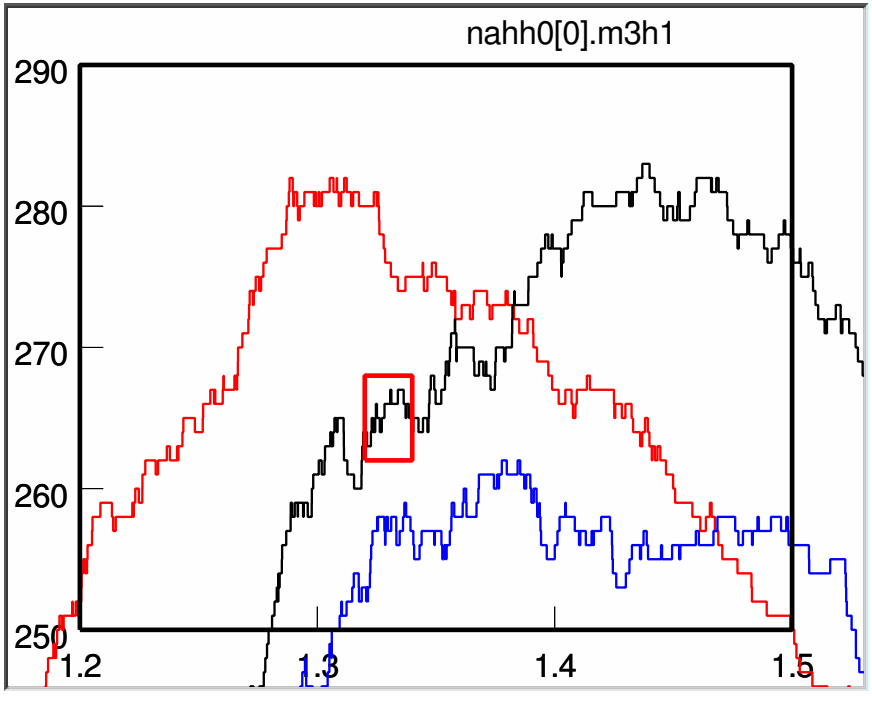
m0h0 <–> m1h0 (a, b) (KSTrans[14])

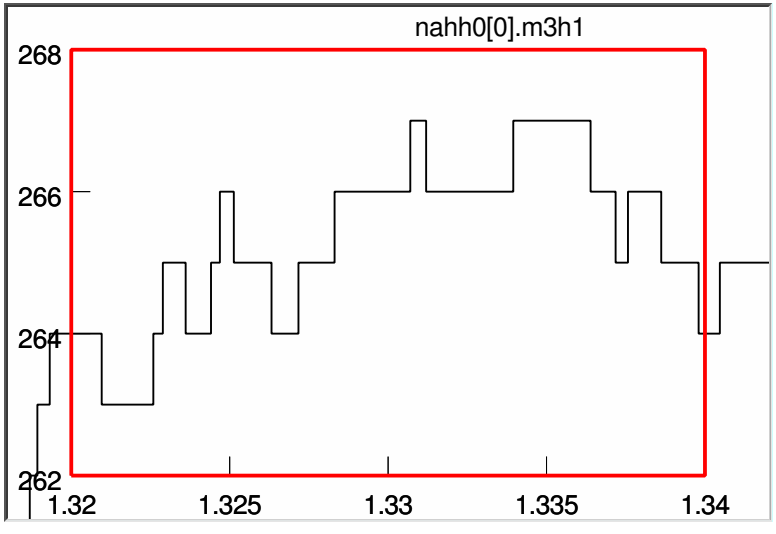
Display inf, tau

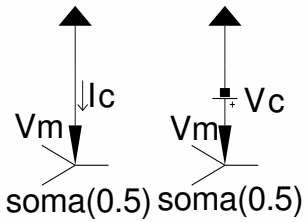
am0h0m1h0 = A*x/(1 – exp(–x)) where x = k*(v – d)

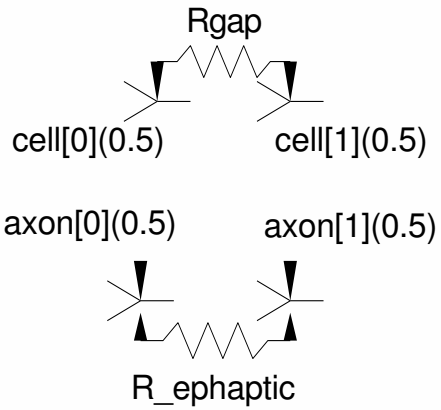

nahh0[0].m3h1
20935 points
300
200
100
0
0
1
2
3
4
5

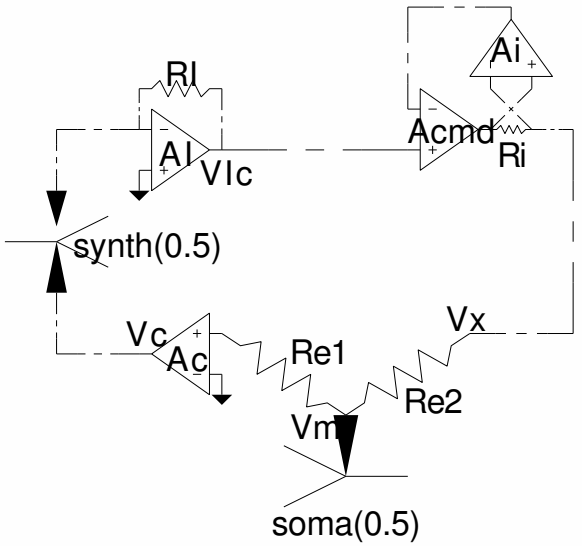

Use variable dt
Absolute Tolerance
0.001
Atol Scale Tool
Details

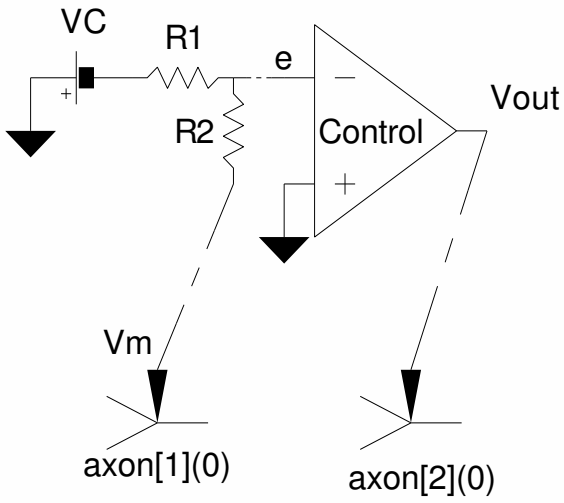

nahh0[0].m3h1
290
280
270
260
250
1.2
1.3
1.4
1.5

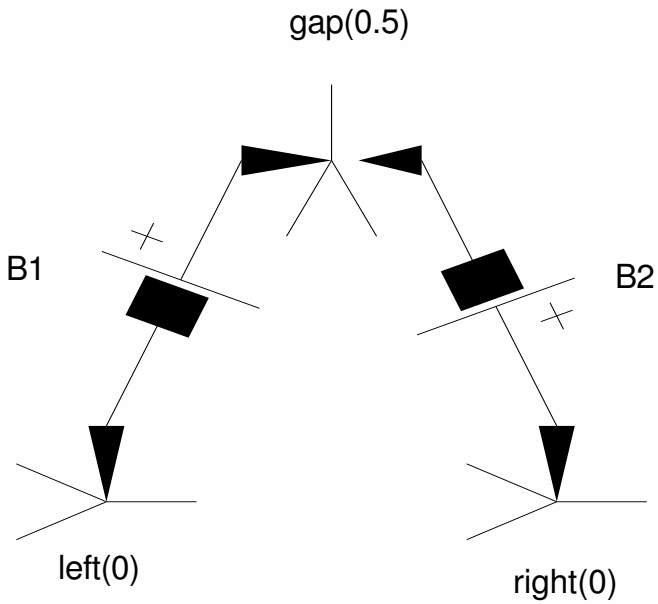

nahh0[0].m3h1
268
266
264
262
1.32
1.325
1.33
1.335
1.34

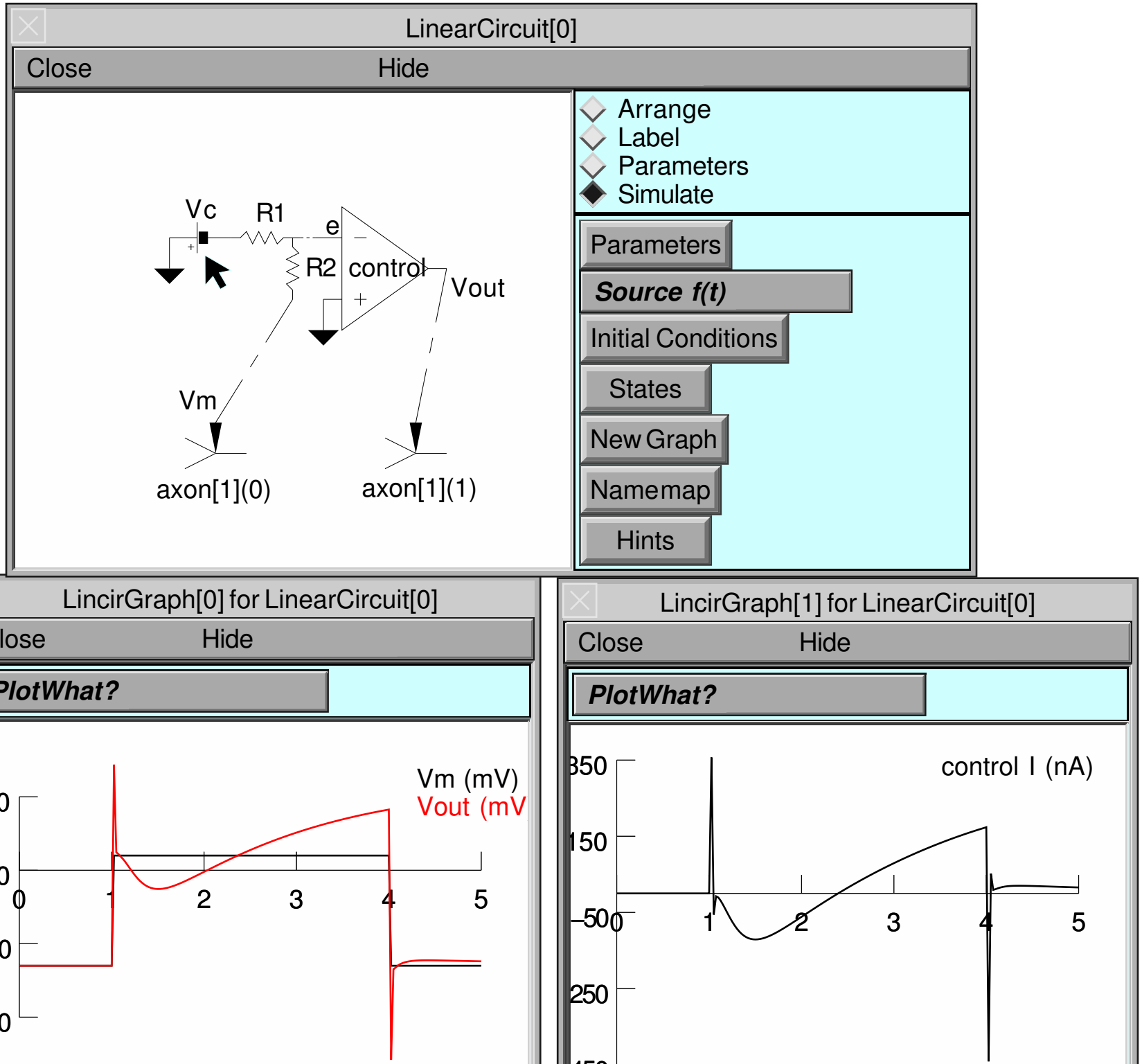

Ic
Vm
soma(0.5)
Vc
Vm
soma(0.5)

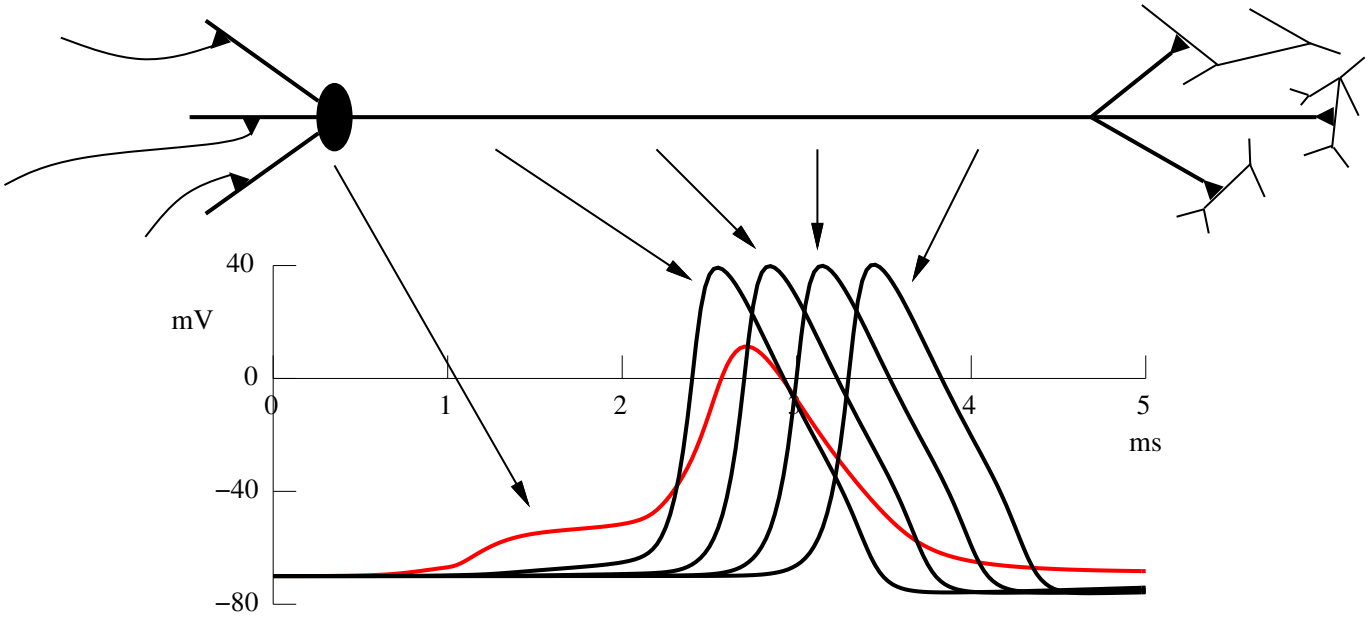

Rgap
cell[0](0.5)
cell[1](0.5)
axon[0](0.5)
axon[1](0.5)
R_ephaptic

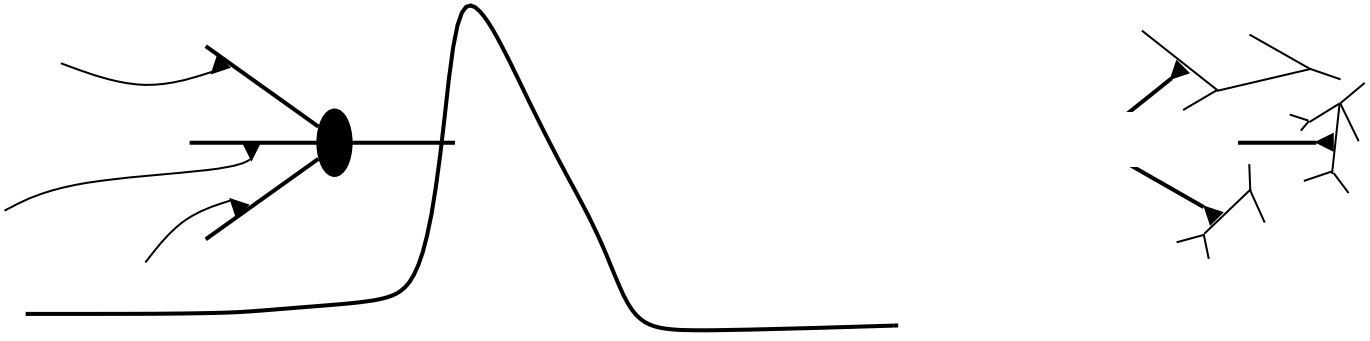

Rl
Ai
Acmd
Ri
Al
Vlc
synth(0.5)
Vc
Ac
Re1
Vx
Re2
Vm
soma(0.5)


VC
R1
e
Vout
R2
Control
Vm
axon[1](0)
axon[2](0)

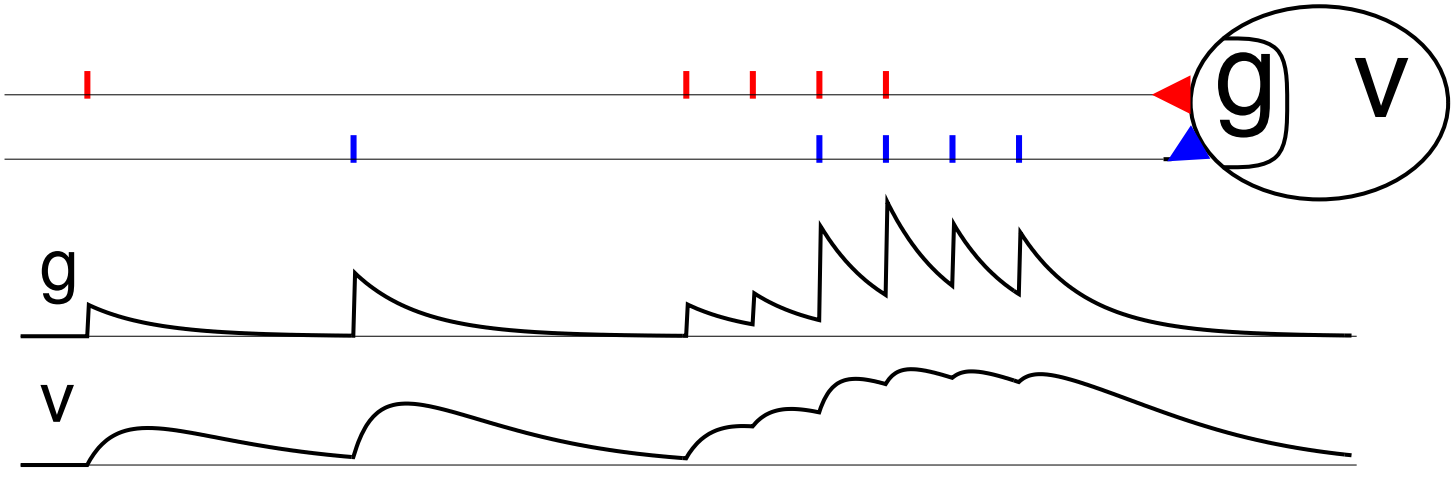

gap(0.5)
B1
B2
left(0)
right(0)


LinearCircuit[0]
Close
Hide
Vc
R1
e
R2
control
Vout
Vm
axon[1](0)
axon[1](1)
Arrange
Label
Parameters
Simulate
Parameters
Source f(t)
Initial Conditions
States
New Graph
Namemap
Hints
LincirGraph[0] for LinearCircuit[0]
Close
Hide
PlotWhat?
Vm (mV)
Vout (mV
50
0
−50
100
0
1
2
3
4
5
LincirGraph[1] for LinearCircuit[0]
Close
Hide
PlotWhat?
control I (nA)
350
150
−50
250
0
1
2
3
4
5



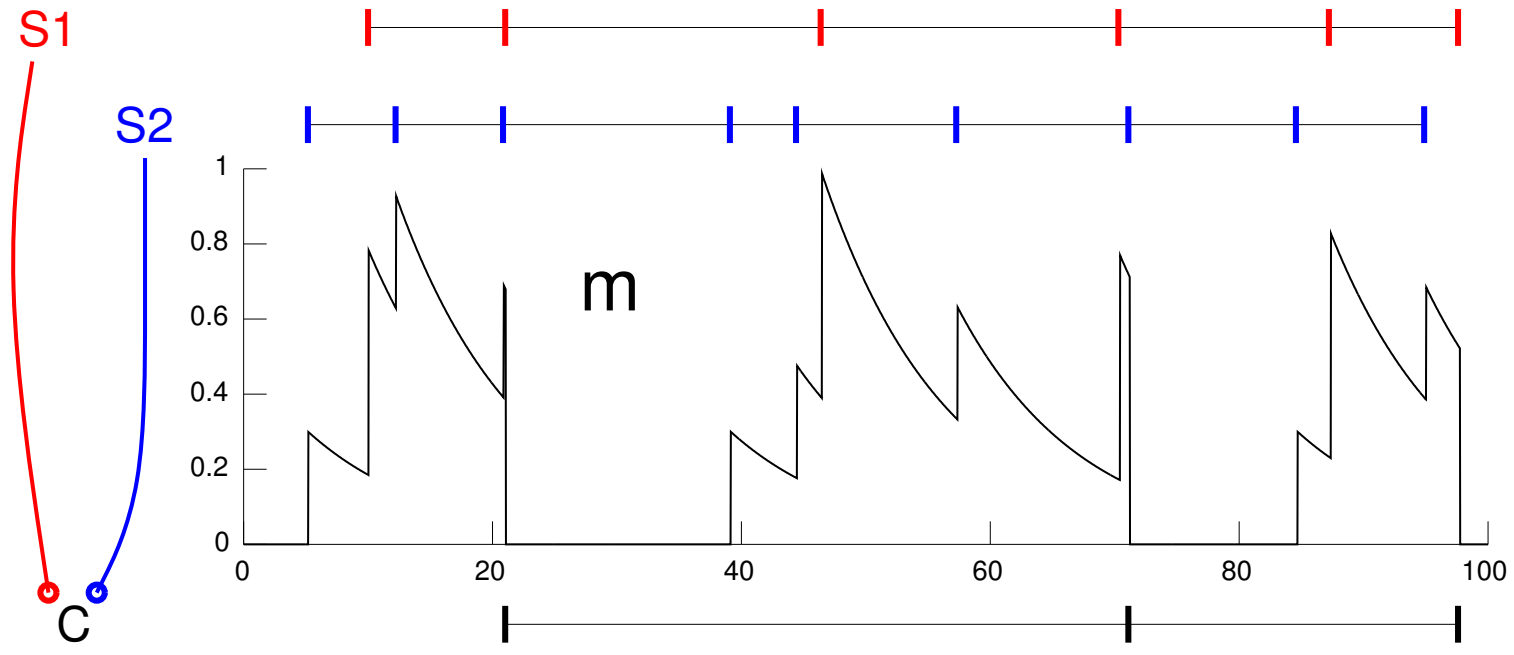


```
                                .threshold
new NetCon(src, target)         .delay
                                .weight
```



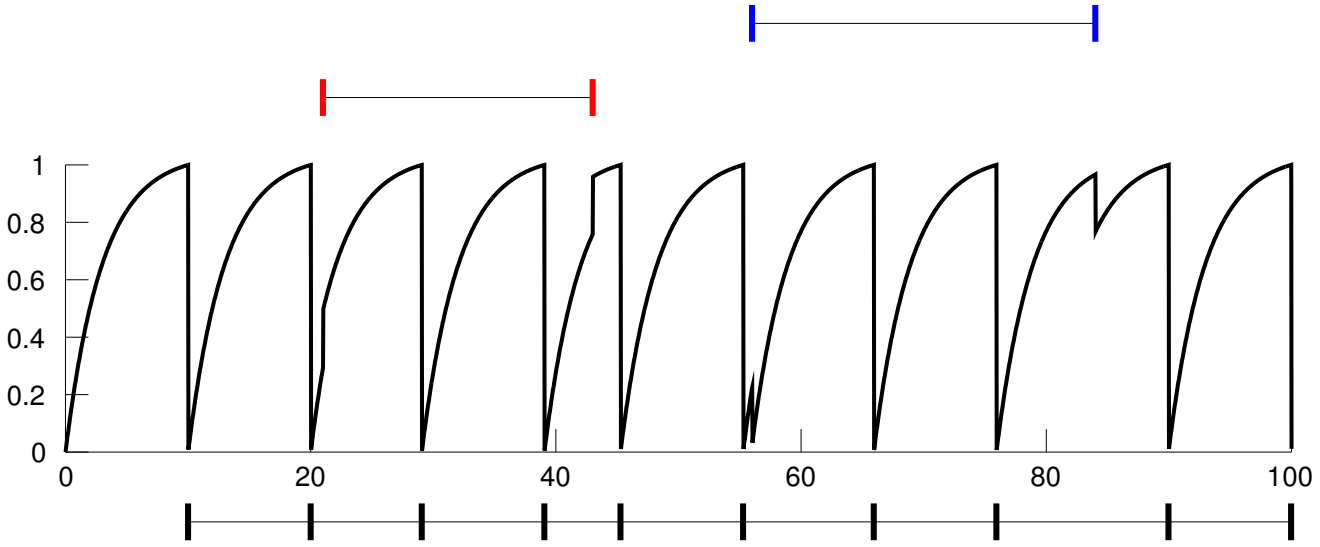

```
NEURON {
  POINT_PROCESS ExpSyn
  RANGE tau, e, i
  NONSPECIFIC_CURRENT i
}

...declarations...

INITIAL { g = 0 }
```

```
CHANNEL {
  SOLVE state METHOD cnexp
  i = g*(v - e)
}

DERIVATIVE state {
  g' = -g/tau
}

NET_RECEIVE(w) {
  g = g + w
}
```



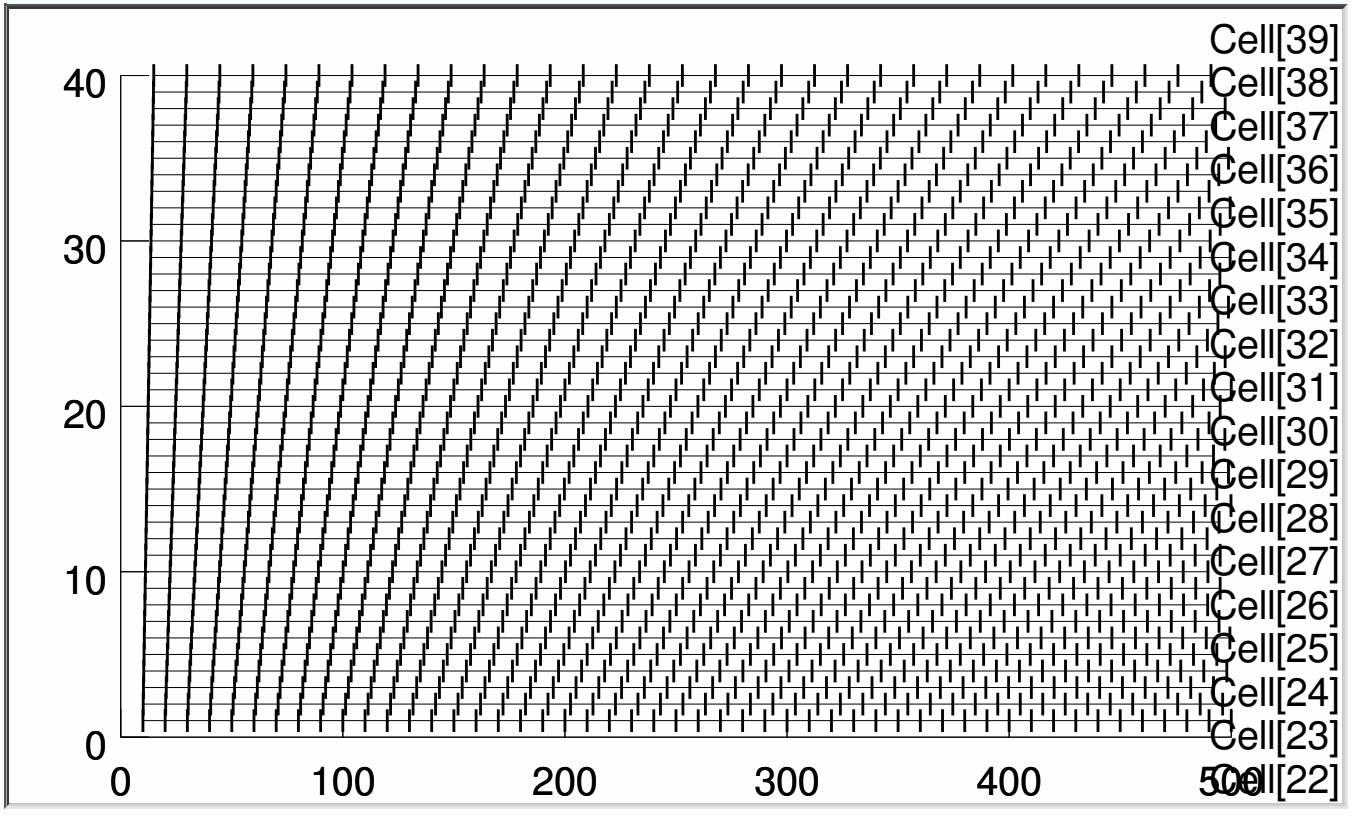

```
NEURON {
  ARTIFICIAL_CELL IntFire
  RANGE tau, m
}

...declarations...
```

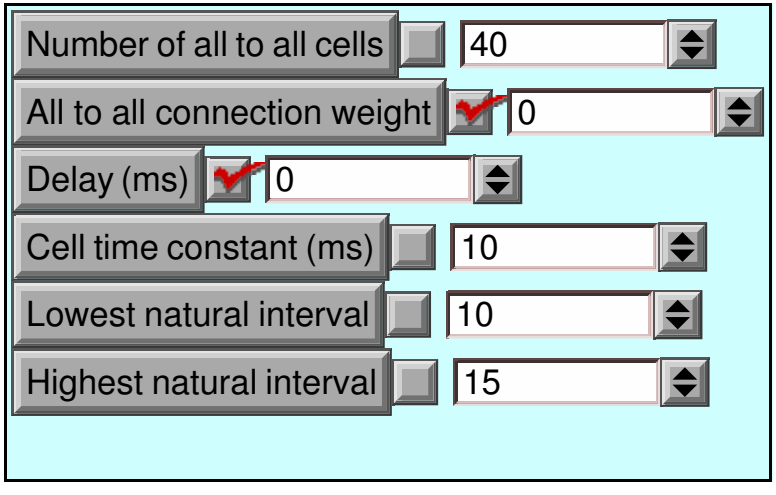
```
INITIAL { m = 0   t0 = t }

NET_RECEIVE (w) {
  m = m*exp(-(t - t0)/tau)
  t0 = t
  m = m + w
  if (m > 1) {
    net_event(t)
    m = 0
  }
}
```

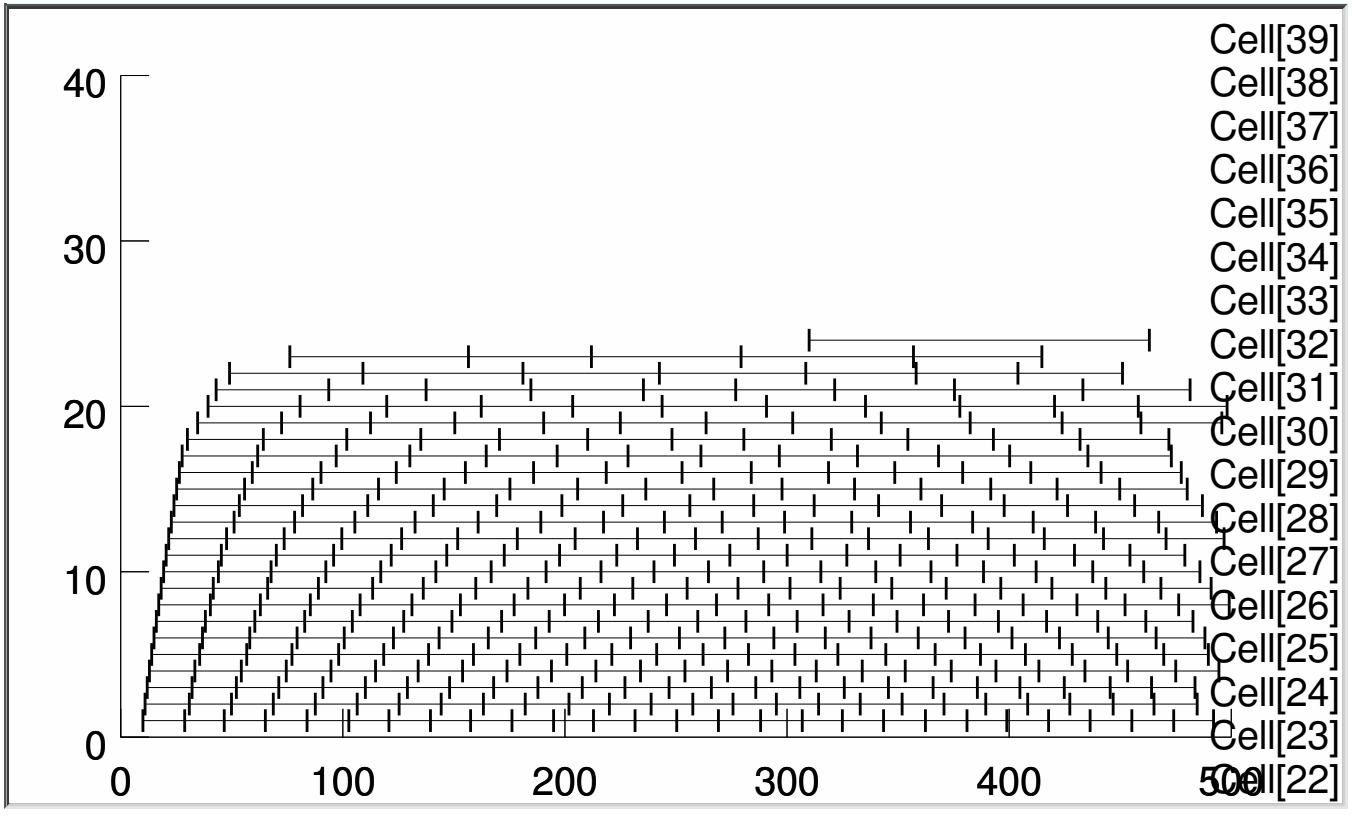


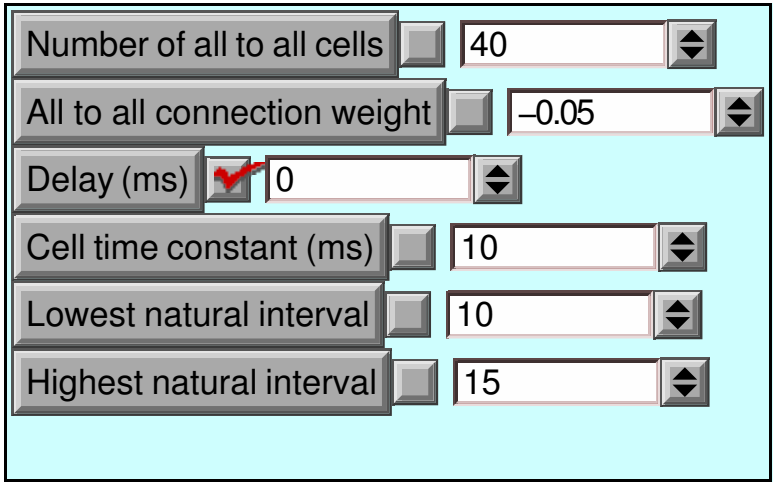

```
: dm/dt = (minf - m)/tau
: input event adds w to m
: when m = 1, or event
:  makes m >= 1, cell fires
: minf is calculated so
:  that the natural interval
:  between spikes is invl

INITIAL {
   minf = 1/(1 - exp(-invl/tau))
   m = 0
   t0 = t
   net_send(firetime(), 1)
}
```

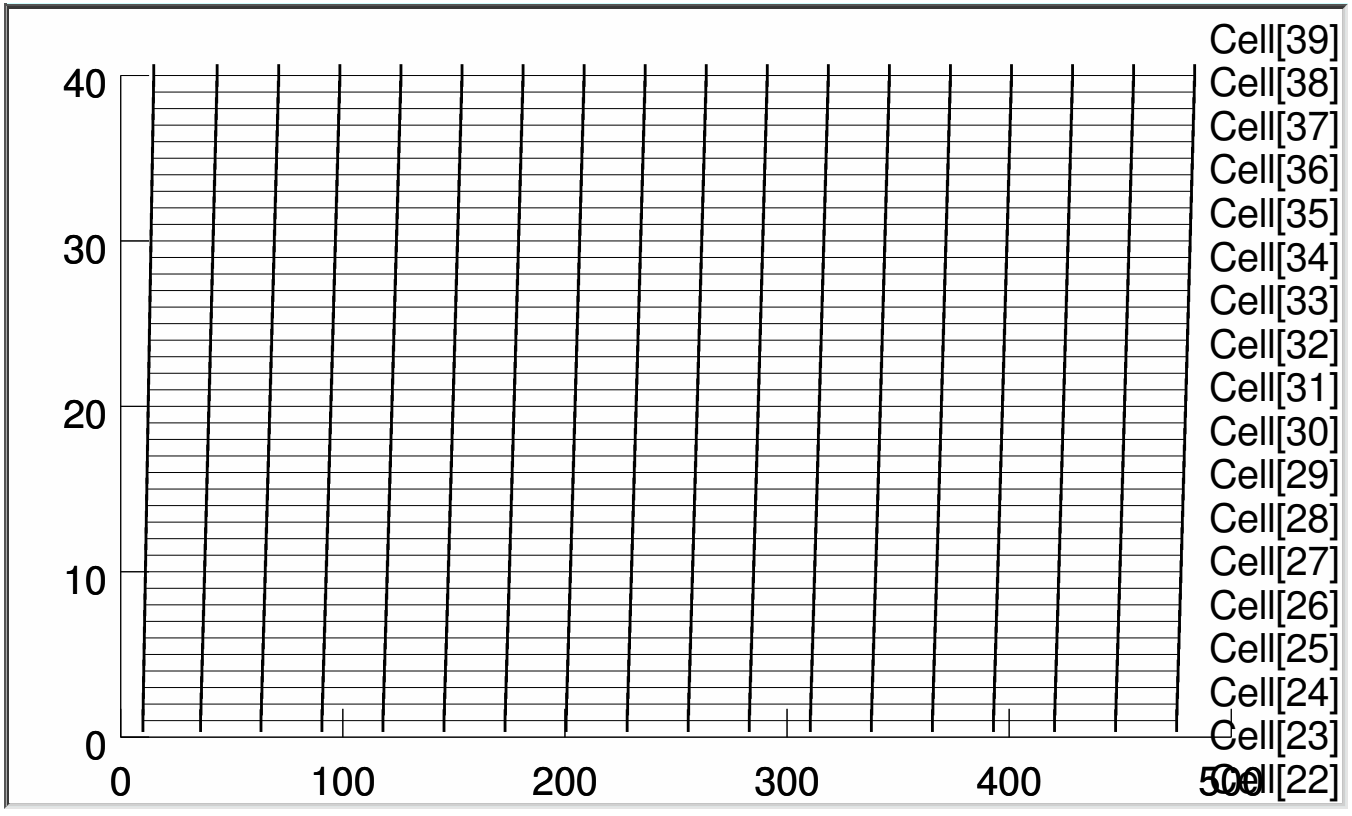
```
NET_RECEIVE (w) {
   m = minf + (m - minf)*exp(-(t - t0)/tau)
   t0 = t
   if (flag == 0) {
      m = m + w
      if (m > 1) {
         m = 0
         net_event(t)
      }
      net_move(t+firetime())
   }else{
      net_event(t)
      m = 0
      net_send(firetime(), 1)
   }
}

FUNCTION firetime() { : m < 1 < minf
   firetime = tau*log((minf-m)/(minf - 1))
}
```

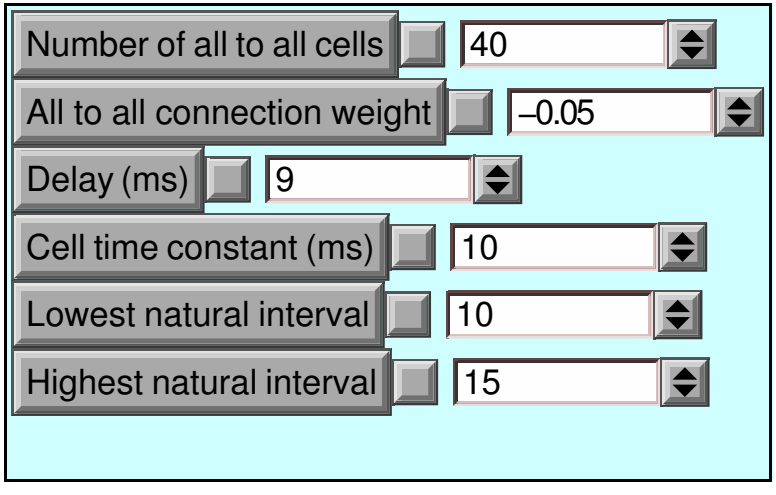

Cell[39]
Cell[38]
Cell[37]
Cell[36]
Cell[35]
Cell[34]
Cell[33]
Cell[32]
Cell[31]
Cell[30]
Cell[29]
Cell[28]
Cell[27]
Cell[26]
Cell[25]
Cell[24]
Cell[23]
Cell[22]
40
30
20
10
0
0
100
200
300
400
500

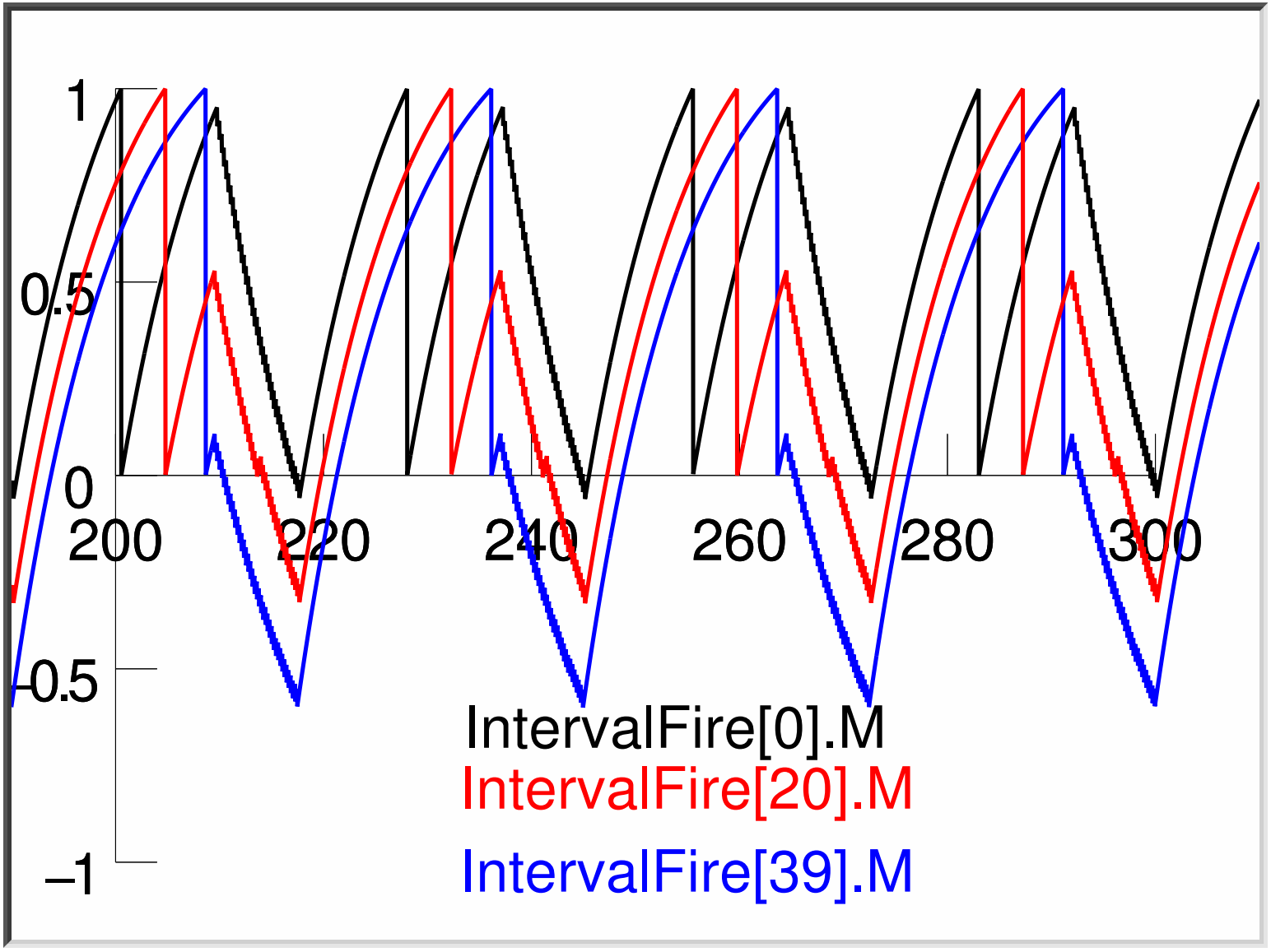

Number of all to all cells
40
All to all connection weight
0
Delay (ms)
0
Cell time constant (ms)
10
Lowest natural interval
10
Highest natural interval
15

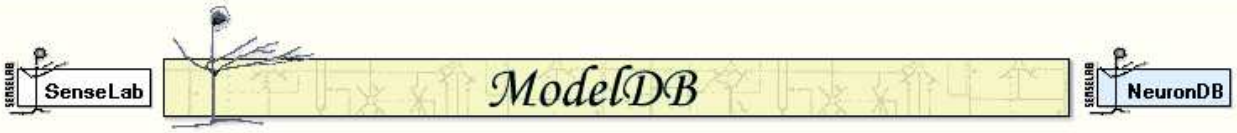

40
30
20
10
0
0
100
200
300
400
500
Cell[39]
Cell[38]
Cell[37]
Cell[36]
Cell[35]
Cell[34]
Cell[33]
Cell[32]
Cell[31]
Cell[30]
Cell[29]
Cell[28]
Cell[27]
Cell[26]
Cell[25]
Cell[24]
Cell[23]
Cell[22]


Number of all to all cells
40
All to all connection weight
–0.05
Delay (ms)
0
Cell time constant (ms)
10
Lowest natural interval
10
Highest natural interval
15


Cell[39]
Cell[38]
Cell[37]
Cell[36]
Cell[35]
Cell[34]
Cell[33]
Cell[32]
Cell[31]
Cell[30]
Cell[29]
Cell[28]
Cell[27]
Cell[26]
Cell[25]
Cell[24]
Cell[23]
Cell[22]
40
30
20
10
0
0
100
200
300
400
500


Number of all to all cells
40
All to all connection weight
–0.05
Delay (ms)
9
Cell time constant (ms)
10
Lowest natural interval
10
Highest natural interval
15


1
0.5
0
200
220
240
260
280
300
–0.5
–1
IntervalFire[0].M
IntervalFire[20].M
IntervalFire[39].M



ModelDB provides an accessible location for storing and efficiently retrieving computational neuron models. ModelDB is tightly coupled with NeuronDB. Models can be coded in any language for any environment, though ModelDB has been initially constructed for use with NEURON and GENESIS. Model code can be viewed before downloading and browsers can be set to auto-launch the models. **Help**

- Search for models by author name Lytton Search

- List models sorted by first author, by each author or by model name
- Find models of a particular Neuron type
- Find models containing a particular Property: Currents, Receptors, or Transmitters
- Find models that relate to a Topic, e.g. synaptic plasticity, pattern recognition, etc.
- Find models that run in a particular Simulation environment
- List models of: Networks, Neurons, Synapses (and ligand-gated ion channels), Neuromuscular Junctions, Axons, voltage-gated Ion Channels
- Find models containing the following words Search
  Case Sensitive

- Search for publications in ModelDB or in PubMed

- Create a private model
- Signup for an account for creating and editing models privately

SenseLab ModelDB

Search for Models by a particular author

Enter an author's name in the format: Huxley AF (initials optional)
Wildcard (*) searching is supported

Lytton Search

Search Results

The following author names match 'Lytton': Lytton WW, Lytton W, Lytton J

Models by Lytton

1. Local variable time step method by Lytton and Hines 2005
   Lytton W, Hines M (2005) Independent variable time-step integration of individual neurons for network simulations. *Neural Computation* **17**
2. Signal integration in LGN cells by Briska et al 2003
   Briska AM, Uhlrich DJ, Lytton WW (2003) Computer model of passive signal integration based on whole-cell in vitro studies of rat lateral geniculate nucleus. *Eur J Neurosci* **17**:1531-41 [PubMed]
3. Hippocampus temporo-septal engram shift model: Lytton 1999
   Lytton WW, Lipton P (1999) Can the hippocampus tell time? The temporo-septal engram shift

Hippocampus temporo-septal engram shift model: Lytton 1999

Temporo-septal engram shift model of hippocampal memory. The model posits that memories gradua hippocampus from a temporal encoding site to ever more septal sites from which they are recalled. W sense of time is encoded by the location of the engram along the temporo-septal axis.
Reference: Lytton WW, Lipton P (1999) Can the hippocampus tell time? The temporo-septal engram *Neuroreport* **10**:2301-6 [PubMed]

Citations Citation Browser

Model Information (*Click on a link to find other models with that property*)

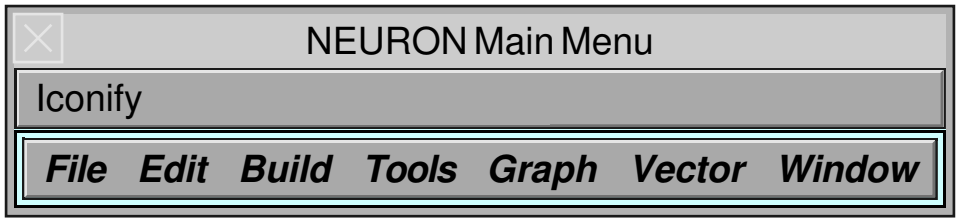
Model Type:	Network;
Cell Type(s):	
Channel(s):	I Na,t; I K;
Receptor(s):	GabaA; AMPA;
Transmitter(s):	
Simulation Environment:	Neuron;
Model Concept(s):	Pattern Recognition; Temporal Pattern Generation; Spatio-temporal Activity Models;
Implementer(s):	Lytton, William ;

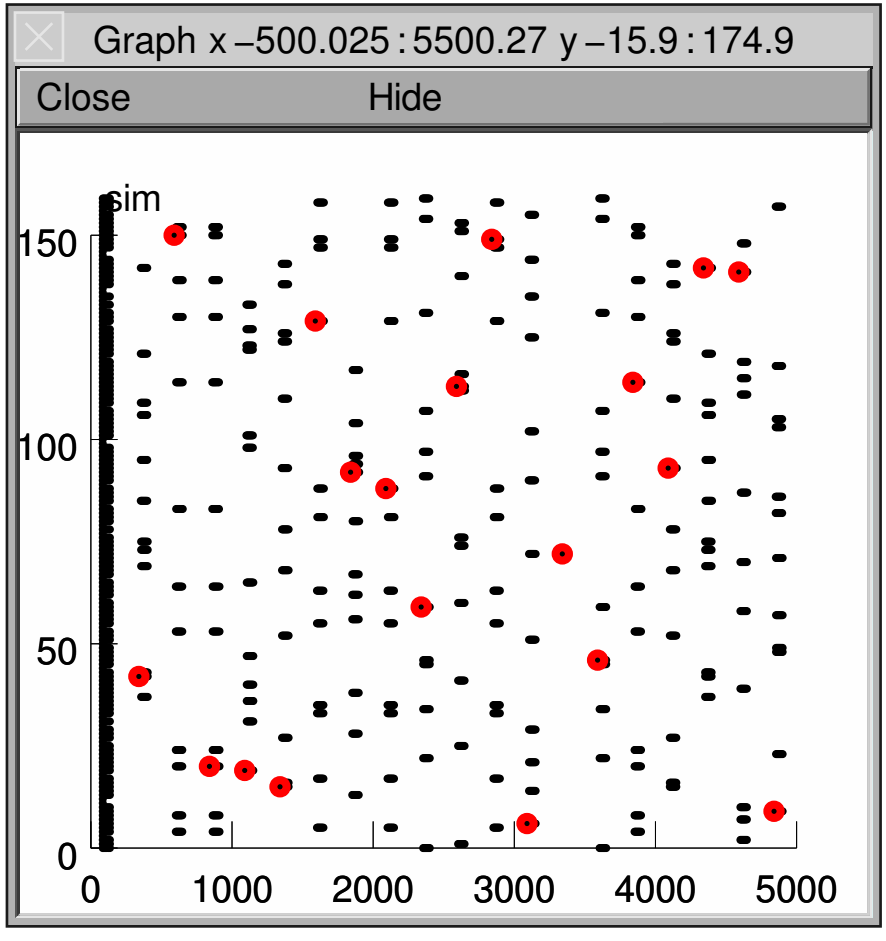
Search NeuronDB for information about: GabaA; AMPA; I Na,t; I K;

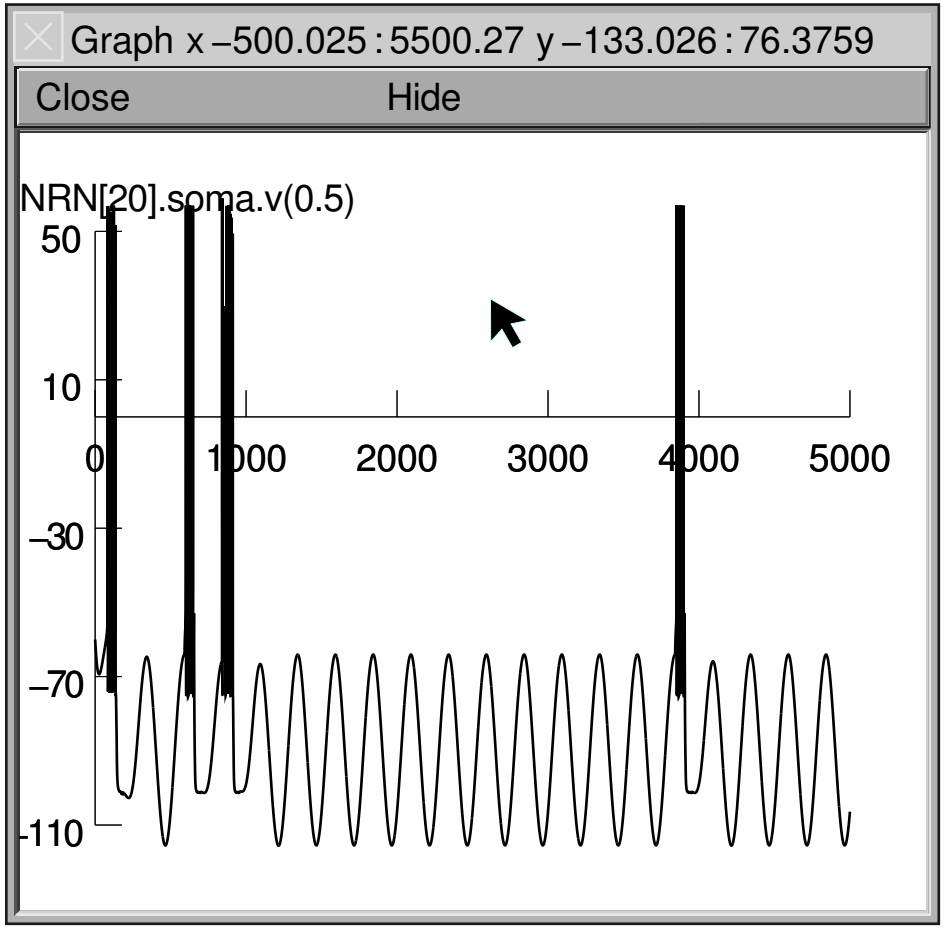
Model files Download zip file | Auto-launch | **Help downloading and running models**

- \
  - lytton99
    - README
    - fig1.gif

```
Lytton WW; Lipton P. Can the hippocampus tell time?: The
temporo-septal engram shift model.
Neuroreport, 10:2301--2306, 1999.

This example produces Fig. 1 p 2302 as shown in fig1.gif.
```


NEURON Main Menu
Iconify
File Edit Build Tools Graph Vector Window


Graph x –500.025 : 5500.27 y –15.9 : 174.9
Close
Hide
sim
150
100
50
0
0
1000
2000
3000
4000
5000


Graph x –500.025 : 5500.27 y –133.026 : 76.3759
Close
Hide
NRN[20].soma.v(0.5)
50
10
–30
–70
-110
0
1000
2000
3000
4000
5000