

The NEURON simulation environment A brief introduction

Michael Rudolph



Advanced Course in Computational Neuroscience Obidos, Portugal, 2004

References

web:

> http://www.neuron.yale.edu/

> http://neuron.duke.edu/

papers:

- Hines, M.L. and Carnevale, N.T. The NEURON Simulation Environment. Neural Computation 9 (1997), 1179-1209.
- Hines, M.L. and Carnevale, N.T. Expanding NEURON's repertoire of mechanisms with NMODL. Neural Computation 12 (2000), 839-851.
- Hines, M.L. and Carnevale, N.T. NEURON: a tool for neuroscientists. The Neuroscientist 7 (2001), 123-135.

The NEURON book (in preparation)

Outline

The idea behind NEURON

- **2** Some mathematics behind NEURON
- **3** Running NEURON
- **4** Two programming levels = infinite possibilities
- **6** A step towards infinity: some examples

What is **NEURON**?

NEURON is a simulation environment for models of biophysical processes, models of individual neurons and networks of neurons.



The idea behind NEURON



The idea behind NEURON

What can NEURON do?

Generally:

NEURON provides tools for constructing, exercising, and managing simplified up to biologically realistic models of electrical and chemical signaling in neurons and networks of neurons.

Specifically:

NEURON can simulate:

 biophysical and biochemical dynamics of active membrane properties (transmembrane currents, transmembrane channels)



[Fundamental Neuroscience. Zigmond et al. (eds.)]

Obiophysical and biochemical dynamics of synaptic transmission



[Fundamental Neuroscience. Zigmond et al. (eds.)]

The idea behind NEURON

electrotonic and active signalling along dendritic and axonal cables, as well as in cable structures with complex branching morphology



А

Vm (mV)

36 m B C D O Electrical Voltage-dependant Chemical conductances (vd) synapses (es sysnapses (cs) C Academic Press items and derived items

copyright © 1999 by Academic Press

The idea behind NEURON

 information processing in simplified up to morphologically and biophysically realistic neuronal models





 information processing in neuronal microcircuits and large networks of interconnected neurons (ranging from simplfied models, such as IAF, up to more realistic neuronal models)







The idea behind NEURON

Moreover:

NEURON provides:

tools for analyzing experimental data





X MulRunFitter[0]	9	and the second second	□ ×
Close	Hide		
Unnamed multiple ru	n proto	col ErrorValue	0.82688
Parameters		Generators	Toggle
A		+ Unnamed Fur	nction Fi 🔟
k1 d1			
k2			
d2	V		V

The idea behind NEURON

 tools for interfacing models in computo with biological neurons in vitro and in vivo (e.g. dynamic clamp)



Why using NEURON?

- (very) easy to use
- suited for computational models on many levels (subcellular, cellular, network)
- well suited for computational models that are closely linked to experimental data
- computationally efficient and accurate while at the same time minimizing the required user effort
- optimized for handling tree-shaped cable structures
- optimized network simulations utilizing the event-based approach
- user is not required to translate the problem into another domain, instead user is able to deal directly with concepts that are familiar at the neuroscience level

The idea behind NEURON

Why not using NEURON?



Hannman, 388

And in the case of questions?

- O Michael Hines michael.hines@yale.edu
- John W. Moore jwm@neuro.duke.edu
- Ted Carnevale ted.carnevale@yale.edu
- ouser group listserv@biomed.med.yale.edu

Outline

The idea behind NEURON

Some mathematics behind NEURON

- **3** Running NEURON
- **4** Two programming levels = infinite possibilities
- **6** A step towards infinity: some examples

NEURON's basic computational task:

to numerically solve the cable equation which describes the relationship between current and voltage in an one-dimensional cable

$$g_{in}\frac{\partial^2 V(t,x)}{\partial x^2} = I_m(t,x) = C\frac{\partial V(t,x)}{\partial t} + g_m V(t,x)$$

V membrane potential $V = V_{in} - V_{out}$

C membrane capacity g_m membrane conductance g_{in} intracellular (axial) conductance

I m membrane current I in intracellular (axial) current



Principal problem of computational modeling:

in reality: space and time are continuous variables and dynamics is described by differential equations

in computo: explicit solution of describing differential equations is often not available

Solution:

approximate continuous system by one which is discontinuous in space and time (dynamics described by difference equations)

compartmentalization (spatial discretization) temporal discretization



compartmental modeling

Compartmental modeling:



compartmentalization (spatial discretization) reduction of cable equation to a set of ordinary differential equations with first order derivatives in time

temporal discretization

algebraic difference equations which can be solved numerically

Methods of numerical integration:

- forward Euler method (simple, unstable, inaccurate)
- backward Euler method (inaccurate, stable)
- Crank-Nicholson method (stable, more accurate)

used in **NEURON**



selection of a method of numerical integration is guided by concerns of stability, accuracy and efficiency

An example for numerical integration: the forward Euler method

basic idea: extrapolation of function V(t) from the value at time t to a new value a brief interval Δt into the future



Outline

The idea behind NEURON

- **2** Some mathematics behind NEURON
- **3** Running NEURON
- **4** Two programming levels = infinite possibilities
- **6** A step towards infinity: some examples

A LINUX environment is the natural habitat of the NEURON simulation environment



O UNIX in 10 minutes



Iisting contents of directories / folders

New 🚮 Shell

list names of files / directories:

ls

detailed list of files / directories:

ls -la



🚳 michael on obidos: /home/michael/MyNeu	roscience/Simulation	ns/Layer6Pyramida - Shell - Konsole 🥤	9 - • ×
Session Edit View Settings Help			
Layer6PyramidalNeuron_51-2a/BasicModel 51-2a_low.geo addStimulation IKd_TraubMiles.mod createMorpholo IKm_GutfreundSegev.mod hs_err_pid9650 INa_TraubMiles.mod i686 StimTrigger1_Ho.mod insertActivePr addRecording.hoc insertPassiveF Layer6PyramidalNeuron_51-2a/BasicModel	l_IClamp> ls n.hoc i ogy.hoc m).log m roperties.hoc m Properties.hoc m l_IClamp>∎	insertSynapticProperties.hoc ain.h ain.hoc winimalAMPA_DestexheSejnowski.mo wltiAMPA_DestexheSejnowski.mod wltiGABAa_DestexheSejnowski.mod	id I
New Shell			

6	michael	on obi	dos: /home	s/michael/	MyNeuroscie	nce/Simulatio	ns/Laye	ar6Pyramida - Shell - Konsole 🧕 💶	>
Se	ssion Edit	View	/ Settings	Help					
Lay	er6Pyram	idalN	leuron_51	-2a/Basi	cModel_IC1	amp> ls -la			-
tot	al 194		_		-	-			
dru	xr-xr-x	3	≋ichael	users	760	2004-07-30	14:53		
dru	жr-жr-ж	9	michael	users	360	2004-07-27	14:02		
-r⊎	-rr	1	Nichael	usens	89286	2003-05-25	11:49	51-2a_low.geo	
-r⊎	-rr	1	≋ichael	usens	1985	2003-05-25	11:49	IKd_TraubMiles.mod	
-ru	-rr	1	Nichael	users	2241	2003-05-25	11:49	IKm_GutfreundSegev.mod	
-ru	-rr	1	≋ichael	usens	2405	2003-05-25	11:49	INa_TraubMiles.mod	
-ru	-rr	1	Nichael	users	1744	2003-05-25	11:49	StimTrigger1_Ho.mod	
-ru	-rr	1	≋ichael	users	2095	2003-05-25	11:52	addRecording.hoc	
-r⊎	-rr	1	michael	users	1874	2003-06-25	20:13	addStimulation.hoc	
-r⊎	-rr	1	michael	usens	2684	2003-05-25	11:49	createMorphology.hoc	
-ru	-rr	1	nichael	users	6639	2003-05-31	18:57	hs_err_pid9650.log	
dru	xr-xr-x	- 3	≋ichael	usens	1416	2003-06-25	19:58	1686	
-ru	-rr	1	≋ichael	users	2583	2003-05-25	11:49	insertActiveProperties.hoc	
-ru	-rr	1	≋ichael	users	2100	2003-05-25	11:49	insertPassiveProperties.hoc	
-r⊎	-rr	1	michael	users	11468	2003-05-25	11:49	insertSynapticProperties.hoc	
-r⊎	-rr	1	Nichael	users	6182	2003-06-01	22:38	main.h	
-r⊎	-rr	1	≋ichael	usens	7613	2003-05-25	11:51	main.hoc	
-r⊎	-rr	1	≋ichael	users	5763	2003-05-25	11:49	minimalAMPA_DestexheSejnowski.mod	
-ru	-rr	1	≋ichael	users	8916	2003-05-25	11:49	multiAMPA_DestexheSejnowski.mod	
-r⊎	-rr	1	∎ichael	users	8912	2003_05-25	11:49	multiGABAa_DestexheSejnowski.mod	
Lay	er6Pyram	idalN	leuron_51	-2a/Basi	cModel_IC1;	amp>			

UNIX in 10 minutes

1 UNIX in 10 minutes

ermission attributes



- Changing directories / folders
 - Change to home directory:
 - Change one directory up:
 - change to specific directory:



Creating new directories / folders

create directory with specific name: mkdir <name>



1 UNIX in 10 minutes

running programs

run programs by typing name into command shell, e.g.

/home/michael> mathematica

/home/michael> nrnmodl

editors

● for hardcore users:

vi

michael on obidos: /home/michael/MyNeuroscience/Simulations/Layer6Pyra	amida - Shell - Kc	×
Session Edit View Settings Help		
// inserting synaptic mechanisms		1
//[SYN print "[SYN if (xopen("insertSynapticProperties.hoc")) {}	APTIC PROPERTIE	- ES]"
// // adding stimulation //		
print " if (xopen("addStimulation.hoc")) {}	[STIMULATIO	"[אכ
// // adding recording //∎		
print " if (xopen("addRecording.hoc")) {}	[RECORDIN	4G]" =
print "====================================		
// -[initialization]- proc init() { finitialize(v_init) fcurrent() if (ENABLE_BACKGROUND) openAll_RNGfiles() "main.hoc" 194L, 7613C	149,3	81%
New Shell		

O UNIX in 10 minutes

O UNIX in 10 minutes

michael on obidos: /home/michael/MyNeuroscience/Simul	ations/Layer6Pyramida - Shell - Ko	
Session Edit View Settings Help		for officient uppers:
IW main.hoc Row 1 print " if (xopen("addRecording.hoc")) {}	77 Col 1 8:37 Ctrl-K H for help 📄	Tor enicient users.
print "====================================		joe
<pre>// simulation procedures //</pre>		
<pre>proc init() { finitialize(v_init) fcurrent()</pre>	🥔 main hoc - KWrite 🔍	
if (ENABLE_BACKGROUND) openAll_RNGfiles() initAll RNGdata()	<u>File Edit Bookmarks Tools Settings Help</u>	
rec_recordVm()		
<pre>// -[single step]- proc step() { local i if (ENABLE_GRAPHICAL_INTERFACE) Plot() if (ENABLE_BACKGROUND) updateAll_RNGdata() for i=1 pstep steprup { advance() }</pre>	print " if (xopen("addRecording.hoc")) {}	[RECORDING]"
rec_recordVm() if (ENABLE_BACKGROUND) resetAll_RNGdata() }	print "====================================	
New Shell	<pre>// -[initialization]-</pre>	
	finitialize(v_init) fcurrent() if (ENABLE_BACKGROUND) openAll_	RNGfiles()
for Windows [©] ers:	initali_RNGdata() rec_recordVm()	
kwrite	<pre>// -[single step]- proc step() { local i if (ENABLE_GRAPHICAL_INTERFACE) if (ENABLE_BACKGROUND) updateAl for i=1,nstep_steprun { advance rec_recordVm()</pre>	Plot() 1_RNGdata() () }
	Line: 160 Col: 13 INS NORM	

NEURON under UNIX



NEURON under UNIX

"customized" versions of NEURON

- Create / edit specific (neuronal) mechanisms (using modl programing language) and NEURON script files (using hoc programing language)
- compile mod1 files using

\$NEURONHOME/i686/bin/nrnivmodl

New 🚮 Shel

	inchaer on obidos. momentichaer/myNeuroscience/simulations/Layero-yrainida - Sheir - Konsole
	Session Edit View Settings Help
 compiles all mod1 files in current directory creates directory ./i686 creates executable ./i686/special ("customized" NEURON version) 	Layer6PyramidalNauron 51-2a/BasicModel_IClamp> /usr/local/share/nrn5.6/i686/bin/nrnivmodl "/usr/local/share/nrn5.6/share/nrn/libtool"wode=compile gcc -DHAVE_CONFIG_H -III"/usr/loca l/share/nrn5.6/include/nrn" -I"/usr/local/share/nrn5.6/i686/lib" -g -O2 -c -o mod_func.lo mod_func. .c .gcc -DHAVE_CONFIG_H -III/usr/local/share/nrn5.6/include/nrn -I/usr/local/share/nrn5.6/i686/lib b -g -O2 -c mod_func.c -fPIC -o .libs/mod_func.o "/usr/local/share/nrn5.6/share/nrn/libtool"wode=link gcc -wodule -g -O2 -o libnrnmech.la -rpath //usr/local/share/nrn5.6/i686/lib" IKd_TraubMiles.lo INa_TraubMiles.lo mod_func.lo -("/usr/local/share/nrn5.6/i686/lib" hare/nrn5.6/i686/lib" -lnrnoc -loc -lmemacs -lscopmath -lsparse13 -lreadline -lncurses -ldl -lm rm -fr .libs/libnrnmech.la .libs/libnrnmech.la .libs/libnrnmech.so .libs/libnrnmech.so.0libs/libn gcc -shared .libs/IKd_TraubMiles.o .libs/INa_TraubMiles.o .libs/mod_func.o -W1,rpath -W1,/usr/loca al/share/nrn5.6/i686/lib -U1,rpath -V1,/usr/local/share/nrn5.6/i686/lib -L/usr/N1186/lib -L/usr/loca al/share/nrn5.6/i686/lib /usr/local/share/nrn5.6/i686/lib/libnrnmech.so .ousr/local/share/nrn5.6/i686/lib /libc.so /usr/local/share/nrn5.6/i686/lib/libnrnmech.so.0.0.0 (cd .libs && rm -f libnrnmech.so .0 && ln -s libnrnmech.so .0.0 libnrnmech.so.0.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.so.0.0.0 libnrnmech.so.0.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.so.0.0.0 libnrnmech.so.0.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.so.0.0.0 libnrnmech.so.0.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.so.0.0.0 libnrnmech.so.0.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.so.0.0.0 libnrnmech.so.0.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.so.0.0.0 libnrnmech.so.0.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.so.0.0.0 libnrnmech.so.0 (cd .libs && rm -f libnrnmech.so && ln -s libnrnmech.la libnrnmech.la) Successfully created i686/special Layer6PyramidalNeuron_51-2a/BasicModel_ICla

2 NEURON under UNIX

run customized version of NEURON

./i686/special -

🐻 michael on obidos: /home/michael/MyNeuroscience/Simulations/Layer6Pyramida - Shell - Konsole 🥯 💶 🗙
Session Edit View Settings Help
Layer6PyramidalNeuron_51-2a/BasicModel_IClamp> ./i686/special NEURON Version 5.6 2004-5-19 23:5:24 Main (81) by John W. Moore, Michael Hines, and Ted Carnevale Duke and Yale University Copyright 2001
loading membrane mechanisms from /home/michael/MyNeuroscience/Simulations/Layer6PyramidalNeuron_51-2a /BasicModel_IClamp/i686/.libs/libnrnmech.so Additional mechanisms from files IKd_TraubMiles.mod INa_TraubMiles.mod oc>
New Shell

Outline

The idea behind NEURON

- **2** Some mathematics behind NEURON
- **8** Running NEURON
- 4 Two programming levels = infinite possibilities
- **6** A step towards infinity: some examples

4 Two programming levels = infinite possibilities



NEURON Interpreter:

- "calculator" optimized for integrating sets of coupled differential equations
- object oriented syntax

1 first programming level: **hoc**

- definition of neuronal models (morphology, membrane properties, stimulation and recording protocols)
- basic I/O operations
- control of simulation
- 🔵 data analysis

2 second programming level: nmod1

- description of biochmical / biophysical processes in terms of kinetic schemes or sets of differential and algebraic equations
- C-like syntax; allows incorporation of routines written in C

Outline

The idea behind NEURON

- **2** Some mathematics behind NEURON
- **8** Running NEURON
- **4** Two programming levels = infinite possibilities
- A step towards infinity: some examples

Hodgkin & Huxley

Hodgkin and Huxley's squid axon





loligo pealei
[www.mbl.edu]

biophysical

system

the giant axon [Hodgkin *et al.*, *J. Physiol.* **116** (1952), 424-448]



membrane potential time
course following a brief
current injection
[Hodgkin et al., J. Physiol.
116 (1952), 424-448]

Hodgkin & Huxley

$$C\frac{dV(t)}{dt} = -I_m(t)$$

$$I_m(t) = g_{Na}(t) (V(t) - E_{Na}) + g_K(t) (V(t) - E_K) + g_L (V(t) - E_L)$$

$$g_{Na}(t) = \overline{g}_{Na} m^3(t) h(t)$$

$$g_K(t) = \overline{g}_K n^4(t)$$

$$\frac{x \quad E_x \ [mV] \quad g_x \ [mS \ cm^2]}{Na \quad 115 \quad 120}$$

$$K \quad -12 \quad 36$$

$$L \quad 10.6 \quad 0.3$$

0 Θ Θ 0 Θ K⁻ \bigcirc inside 🕞 🕒 outside \oplus + (+)Na⁺ (+ \oplus \oplus (+(+)



Hodgkin & Huxley

$$\frac{dm(t)}{dt} = \alpha_m(V) \left(1 - m(t)\right) - \beta_m(V) m(t)$$

$$\frac{dh(t)}{dt} = \alpha_h(V) \left(1 - h(t)\right) - \beta_h(V) h(t)$$

$$\frac{dn(t)}{dt} = \alpha_n(V) \left(1 - n(t)\right) - \beta_n(V) n(t)$$

Hodgkin-Huxley model
[Hodgkin, Huxley, Katz (19)]

conceptual model





[Hodgkin & Huxley, J. Physiol. 117 (1952), 500-544]



morphology

create axon	create and access named section
access axon	(or compartment)
axon nseg = 100 axon diam = 500 axon L = 40000	define morphological properties and spatial discretization



5	A step towards infinity: sor	ne examples 0 Ho	dgkin & Huxley
	merge everything togethe	er	
	<pre>load_file("nrngui.hoc")</pre>	load graphical user interface	e computational
	create axon access axon axon nseg = 100 axon diam = 500	morphology	NEURON
	axon L = 40000		
	axon insert pas axon Ra = 123.0	passive properties	
	axon insert hh	active properties	
	<pre>objectvar stim stim = new Iclamp(0.5) stim.del = 1 stim.dur = 1 stim.amp = 2000</pre>	stimulation	
	tstart = 0 tstop = 10	global simulation settings	

🔵 ... run it ...

oc> load_file("HHexample.hoc")

• ... change some GUI settings ...

Simulation control menu

X NEURON Main	Menu	9		- 🗆 × 🗋
Iconify				
File Edit Build	Tools	Graph	Vector	Window
	RunCo	ntrol		
	RunBut	tton		R
	VariableStepControl			-
	Point Processes			L
	Distributed Mechanisms			ř.
	Fitting			
	Impeda	ance		
	Model View			
	Miscel	laneous		

X RunControl	9	×)
Close	Hide	
Init (mV) ←	-65	
Init & Run		
Stop		
Continue til (ms)) 🕶 🗾 🕤	\$
Continue for (ms	5) 🛩 📃 1	•
Single Step		
t(ms) 0		
Tstop (ms)	10	•
dt (ms) 📃	0.025	•
Points plotted/m	is 🗾 40	•
Quiet		
Real Time (s))	



Hodgkin & Huxley

6 A step towards infinity: some examples Hodgkin & Huxley O membrane potential time course Graph x-1:11 y-92:52 9 computational Close Hide X NEURON Main Menu ٩ model Iconify 40 r v(.5) NEURON File Edit Build Tools Graph Vector Window Voltage axis 0 6 8 10 Current axis State axis Shape plot -40 Graph Color/Brush x -1 : 11 y -92 : 52 9 Vector movie Hide Close Phase Plane Grapher 40 r -80 v(.5) axon.v(0) axon.v(1) 0 2 4 6 8 10 🗙 Graph x-1:11 y-92:52 🌏 Hide Close -40 40 r View v(.5) nrniv 🌏 Crosshair Variable to graph 0 Plot what? Enter Symbol name: 2 6 8 4 -80 Pick Vector axon.v(0) Color/Brush Show 40 Axis Type Keep Lines ik(0.005) axon. Family Label? il hh(0.005) -80 ina(0.005) Erase ki(0.005 Move Text ko(0.005) Change Text m hh(0.005) n hh(0.005) Delete nai(0.005 nao(0.005) v(0.005) Cancel Accept +





time at various sites along the axon

21		
close	Hide	
40		
30		
20		
10		
10		
20		
20	And a second	
30		
40		
50		
60		
70		
80		

membrane potential along the axon



propagating axonal action potentials



5 A step towards infinity: some examples 2 5-compartment model

2 5-compartment model with Hodgkin-Huxley channels in soma







membrane potential

shape plot

3 custom channels

3 5-compartment model with custom ion channels







membrane potential

shape plot



Description of the point-conductance

point-conductance model of synaptic activity



1000

0

0000

E700

7000

0500

effective stochastic processes for synaptic conductances $g_i(t) \sim f(t) = g_e(t) \qquad f(t) = g_e$

membrane potential

inhibitory synaptic conductance

excitatory synaptic conductance

6 detailed model





1 IAF neural net

Onetwork of randomly connected IAF neurons



spike time analysis

- 🗆 ×

9 spike train analysis

_ _ ×

9 statistical analysis and fitting of spike trains



spike time analysis <2>

⁽¹⁾ X-cells

Dependence of the provide the provide the provide the provided the



parameter menu X-cell r

X-cell response