# BIOGRAPHICAL SKETCH

Provide the following information for the Senior/key personnel and other significant contributors.
Follow this format for each person. DO NOT EXCEED FIVE PAGES.

NAME: HINES, MICHAEL L

eRA COMMONS USER NAME (credential, e.g., agency login): MHINES

POSITION TITLE: Senior Research Scientist

EDUCATION/TRAINING *(Begin with baccalaureate or other initial professional education, such as nursing, include postdoctoral training and residency training if applicable.)*

| INSTITUTION AND LOCATION | DEGREE (if applicable) | Completion Date MM/YYYY | FIELD OF STUDY |
|---|---|---|---|
| Michigan State University | BS | 01/1970 | Physics |
| University of Chicago | MOTH | 01/1972 | Physics |
| University of Chicago | PHD | 01/1975 | Physics |
| University of Chicago, Chicago, Illinois | Postdoctoral Fellow | 04/1976 | Biophysics and Theoretical Biology |

## A. Personal Statement

My career is focused on the development of the NEURON Simulation Environment as a framework for understanding the contributions of anatomy and biophysical properties to the function of neurons and neural circuits. Conceptual control of simulations by experimentalists requires that underlying powerful solver techniques from computational engineering, physics, and chemistry be brought into use in the context of neurobiology domain specific representations. That the NEURON code base has allowed me to do this incrementally in regard to Python as an alternative interpreter, variable time step methods for both ordinary and differential-algebraic equations, and MPI and thread parallelization, to list a few of many examples, testifies to its organizational flexibility and modularity. Since 2006, the transformation of NEURON from a serial to a parallel program with unique capabilities in the areas of distributing individual cells across processors, exploiting Blue Gene/Q hardware support for spike exchange overlapping the computation, and thread parallelization, demonstrates my expertise with regard to MPI and thread programming and my understanding of the benefits of careful attention to cache efficient memory layout. At the same time, maximal usefulness of NEURON makes it essential to enter into collaboration with experimental neuroscience researchers carrying out real computational neuroscience simulation projects in order to help focus my developer attention on present needs.

1. Lytton WW, Hines ML. Independent variable time-step integration of individual neurons for network simulations. Neural Comput. 2005 Apr;17(4):903-21. PubMed PMID: 15829094; PubMed Central PMCID: PMC2712447.

2. Migliore M, Cannia C, Lytton WW, Markram H, Hines ML. Parallel network simulations with NEURON. J Comput Neurosci. 2006 Oct;21(2):119-29. PubMed PMID: 16732488; PubMed Central PMCID: PMC2655137.

3. Hines ML, Markram H, Schürmann F. Fully implicit parallel simulation of single neurons. J Comput Neurosci. 2008 Dec;25(3):439-48. PubMed PMID: 18379867; PubMed Central PMCID: PMC2760991.

4. Hines ML, Davison AP, Muller E. NEURON and Python. Front Neuroinform. 2009;3:1. PubMed PMID: 19198661; PubMed Central PMCID: PMC2636686.

## B. Positions and Honors

### Positions and Employment

| | |
|---|---|
| 1976 - 1993 | Medical Research Associate Professor, Duke University, Durham, NC |
| 1993 - 2012 | Research Scientist, YALE UNIVERSITY |
| 2012 - | Senior Research Scientist, YALE UNIVERSITY |

### Other Experience and Professional Memberships

| | |
|---|---|
| - | Member, Society for Neuroscience |

### Honors

## C. Contribution to Science

1. Neural simulation technology -- Computing speed --

Neural simulation dates back to numerical integrations performed by teams of people sitting at calculators under the direction of Allan Hodgkin and Andrew Huxley. Since that time, computing has of course become much easier, and the speed, scope and ease of neural simulation has vastly increased. Work on these 3 aspects -- speed, scope, ease -- have been three major themes of my NIH-funded research, "Computer Methods for Physiological Problems", now in year 40. My initial simulator for public use, developed in the early 1980s, was called CABLE. It was a tool for simulating unbranched and branched dendrites and axons. An algorithmic innovation that I introduced at that time was a staggered implicit step method that was second order correct in space and time for Hodgkin-Huxley style branched cable equations. This second order correctness was preserved without the need for iterations to handle the gating state and conductance nonlinearities. The method used optimal direct Gaussian elimination by ordering the current balance equations in a way that used exactly the same number of arithmetic operations for branched tree cables as for a single unbranched cable with the same number of equations, greatly improving numerical calculation efficiency and speed (ref a,b). While similar tools had been used in other fields, this represented the introduction of these techniques in neuroscience. Subsequently, in the early 90's, the notions of neuron model description and numerical discretization became completely separated and CABLE became NEURON. NEURON's code base has demonstrated its organizational flexibility and modularity through the incremental addition of Python as an alternative interpreter, variable time step methods for both ordinary and differential-algebraic equations, spike and gap junction coupled network description and simulation, and MPI and thread parallelization, to list a few of many examples.

Recently, we have focused much of our efforts on developing new approaches to parallelization of calculation across multiple processors for use with modern high performance computer clusters. Much of this work has been done in collaboration with the Human Brain Project in Geneva, Switzerland. In a series of papers, we demonstrated NEURON's use to split single cells across multiple processors (multisplit method, ref c), and to optimize interprocessor communication across neurons for very large neuronal networks on one of the largest current HPC systems (ref d). Another recent innovation has been the porting of NEURON to Graphics Processing Units (GPUs) and Intel's Many Integrated Core architecture (MIC) in a manner that exhibits good cache efficiency.


a. Hines M. Efficient computation of branched nerve equations. Int J Biomed Comput. 1984 Jan-Feb;15(1):69-76. PubMed PMID: 6698635.

b. Hines M. A program for simulation of nerve equations with branching geometries. Int J Biomed Comput. 1989 Mar;24(1):55-68. PubMed PMID: 2714879.

c. Hines ML, Markram H, Schürmann F. Fully implicit parallel simulation of single neurons. J Comput Neurosci. 2008 Dec;25(3):439-48. PubMed PMID: 18379867; PubMed Central PMCID: PMC2760991.

d. Hines M, Kumar S, Schürmann F. Comparison of neuronal spike exchange methods on a Blue Gene/P supercomputer. Front Comput Neurosci. 2011;5:49. PubMed PMID: 22121345; PubMed Central PMCID: PMC3219917.

2. Neural simulation scope -- Multiscale modeling --

Interaction of dynamics across levels of organization has been a central theme since the early days of the field, being discussed in the 1988 Science paper entitled 'Computational Neuroscience' that introduced the field's name. The NEURON simulator has gradually grown to encompass multiple levels or scales, and provides a tool for what is now referred to as 'multiscale modeling.' The major change in scope was when CABLE became NEURON, to simulate full anatomically-detailed neurons traced from microscopes (first Eutectics then Neurolucida and others). At about the same time, NEURON was extended to be able to handle networks of neurons. Also early on, I implemented the model description language translator for the National Biomedical Simulation Resource and which I subsequently modfied to become the NMODL language for definition of membrane-associated ion channels, as well as cytoplasmic reaction schemes (ref a.) Recently, the reaction-diffusion extension (this proposal) had added full-featured simulation tools to allow more complete molecular signaling simulation.

NEURON is a general neural simulator and as such is complementary to specialized simulators, such as NEST or BRIAN (at the network scale) or MCELL, STEPS, NeuroRD (at the molecular scale). Each of these specialized simulators offer particular areas of expertise and efficiency. Therefore, we have developed over the years a series of connector programs to allow co-simulation. At the molecular level, these have been developed in limited fashion with both MCELL and STEPS, and are being extended to allow more full-featured connectivity. Additionally, we have provided the ability to import simulations written in SBML (Systems Biology Markup Language). At the network level, we have worked with the simulation agglutinators PyNN and NeuroML to provide the ability to co-simulate in that domain.

a. Hines ML, Carnevale NT. Expanding NEURON's repertoire of mechanisms with NMODL. Neural Comput. 2000 May;12(5):995-1007. PubMed PMID: 10905805.

3. Neural simulation ease of use -- Tool accessibility and dissemination --

As mentioned above, neural simulation has come far from the era of hand calculators. The original CABLE simulator was implemented on a PDP-11. Porting to newer platforms has required the development of a series of more than 20 different installers over the years as the 3 major platforms that have come to dominate computing -- Unix/Linux, DOS/Windows, MacOS – have each upgraded and changed hardware and software many times over the years. We have remained committed to maintaining functionality across all of these platforms, and often across multiple variant operating systems available on 1 platform at a given time.

An early innovation that greatly improved ease of use was the introduction of an interpreted language as a front-end to the simulator. This was originally called hoc (adapted from a Kernighan language of that name) and became oc – object-oriented hoc. This adoption represented a considerable step-up in flexibility of use compared to the use of simple configuration files for simulation organization. Subsequently, the addition of a full-featured Graphical User Interface (GUI) permitted use by many researchers without programming skills. This made the tool available to neurophysiologists who might otherwise not have attempted to add simulation to their repertoire of techniques. More recently, after a series of international conferences on interoperability of neural simulation tools, NEURON has become fully compatible with the Python programming language (ref b).

Another effort that has added considerably to useability has been the ModelDB (and SimtoolDB) database of models under the auspices of the SenseLab project at Yale, which I have been actively involved in since its inception (ref a). ModelDB hosts the program code for close to 1000 published computational neuroscience models from a wide variety of simulators. About half of the simulations are for NEURON and this large variety of simulations means that most new simulation projects are now initiated by drawing pieces from prior simulations rather than starting entirely from scratch. Recently we have developed ModelView (ref c), as well as a series of other browser-based query tools, to allow a user to rapidly see what features and components are present in a particular model.

A large part of getting any software to be used involves teaching people how to use it. The development of teaching aids and tutorials has been a focus of the NEURON project from early on. I coauthored with Ted Carnevale a book explaining the tool. Starting at the beginning of the internet era, we have developed (and maintained and updated) a variety of web sites associated with NEURON source code, versioning systems, tutorials, documentation, a forum and other support tools. Another large part of system support has been answering user queries, feature requests and bug reports.


a. Crasto CJ, Marenco LN, Liu N, Morse TM, Cheung KH, Lai PC, Bahl G, Masiar P, Lam HY, Lim E, Chen H, Nadkarni P, Migliore M, Miller PL, Shepherd GM. SenseLab: new developments in disseminating neuroscience information. Brief Bioinform. 2007 May;8(3):150-62. PubMed PMID: 17510162; PubMed Central PMCID: PMC2756159.

b. Hines ML, Davison AP, Muller E. NEURON and Python. Front Neuroinform. 2009;3:1. PubMed PMID: 19198661; PubMed Central PMCID: PMC2636686.

c. McDougal RA, Morse TM, Hines ML, Shepherd GM. ModelView for ModelDB: Online Presentation of Model Structure. Neuroinformatics. 2015 Oct;13(4):459-70. PubMed PMID: 25896640; PubMed Central PMCID: PMC4618280.


Complete List of Published Work in My Bibliography:
http://1.usa.gov/1MAb5xo

## D. Additional Information: Research Support and/or Scholastic Performance

### Ongoing Research Support

R01 DC009977-07
HINES, MICHAEL L (PI)
08/01/09-07/31/19
SenseLab: Integration of Multidisciplinary Sensory Data
Role: PI

R01 NS011613-40
HINES, MICHAEL L (PI)
07/01/78-12/31/17
Computer Methods for Physiological Problems
Role: PI

R01 MH086638-05
HINES, MICHAEL L (PI)
06/01/10-02/28/17
Extension of NEURON simulator for simulation of reaction-diffusion in neurons
Role: PI

R01 EB022889-01
JONES, STEPHANIE (Brown University)

10/01/16 – 9/30/19
Human Neocortical Neurosolver
Role: CoPI

R01 EB022903-01
Antic/Lytton (SUNY)
10/01/16 – 9/30/19
Embedded-Ensemble Encoding
Role: Key Personnel