Vilnius University
Faculty of Medicine

Neuron simulation

With examples and comparison of selected tools

Yaroslav Rozdobudko

Project for Neurobiology class

November 5, 2023

Contents

1 Introduction

Simulation is an important and useful tool in studying scientific phenomena. More so this is evident in the field of Neuroscience. The complex structure of a brain and the processes that take place inside it produce challenges for scientists. Multiyear and multidisciplinary study (Markram et al. 2015) showed the complexity needed for neuron simulation. Study shows that there are 55 morphological profiles, and 11 electrical types, which in combination gives 207 morpho-electrical types. There might be thousands of connections per neuron, and the behavior of each connection can be different.

So what is meant by neuron simulation? In general, simulation is a some kind of model, where you have an input and an output. We can study provided outputs, update our inputs and or model, and compare the results of our outputs to results provided by experiments. This approach, which is done in combination with experiments, allow scientist to study different phenomena in much more depth, in some cases, it can predict unknown and prove predictions.

In the case of Neuroscience, one of the earliest examples of the model used to study the behavior of neurons was the Hodgkin–Huxley model (Hodgkin et al. 1952), which is a set of differential equations that model how action potentials happen in neurons. This model is an example of a spiking neuron model since there are spikes when neurons are stimulated. While the Hodgkin–Huxley model is accurate and impressive, which gained a Nobel prize for its creators, there were several improvements for this model over the years.

With the invention of computers and the wide usage of computational resources in science new field was developed: Computational neuronscience. This led to the creation of software for the simulation of neurons.

The earliest examples of such software were GENESIS (The General Neural Simulation System) (Wilson et al. 1989) first release 1988, and NEURON (Hines 1993) first release 1993. Letter on more software for simulation of neurons was developed, such as Nest (Gewaltig et al. 2007), which while sacrificing complexity, can be used to simulate bigger and more complex networks, due to being less computationally intensive. A good overview of such software is provided in (Brette et al. 2007).

Over the years there have been a great number and variety of projects that utilize software for neuron simulation. Open worm project (Szigeti et al. 2014) which aims at simulating *Caenorhabditis elegans*, simulates not only neuronal tissue but all tissues and all aspects of the worm. The project is open source and has shown results, one of which was using a real-life robot to simulate the behavior of a worm. Another project is Virtual Fly Brain, which is more of a solution to explore Fly Brain. Provides detailed visualization and information about the fly brain.

Blue Brain Project part of Human Brain Project (Suryawanshi et al. 2013) aims to further develop computational neuroscience. The main goal is to completely digitally reconstruct the human brain. Part of this reconstruction was achieved (Markram et al. 2015) where simulations of a part of a neocortex of a human brain were created. Under the umbrella of this project, multiple software tools were developed.

This project aims to provide an overview of the modeling of neurons, provide a short summary of 3 selected computational simulators: NEURON, Nest, and Brain2, and show a comparison between these tools.

2 History of Neuron modeling

The main focus of simulation in many models is the action potential of a neuron. First was measured in frog nerve-muscle in (Bois-Reymond 1843). Cajal just by looking at the morphology of neurons envisioned that there is some signal between neurons. He noticed spines on dendrites and speculated this is the place where connection may be made. The first attempt to create a mathematical model was made by (Lapicque 1907). It was a simple integrate and fire model that models membrane voltage that changes in time with current stimulation. Letter Hodgkin and Huxley (Hodgkin et al. 1952) developed a complicated model that accurately predicts action potential in response to current. They began their work in 1939 and letter from 1946 to 1952. Development in technology created small enough electrodes that could be inserted in the neuron, however, it was still too big for average neurons, so squid giant axon was used, since it was wide enough for electrodes at that time to be inserted.

Another key part was developing a voltage-clamp in (1947-1949) to control voltage across the membrane. The model was extended further since but remains the main model. A simplified version of the Hudgkin-Huxley model was developed FitzHugh-Nagimo model (FitzHugh 1961) (Nagumo et al. 1962), often used for educational purposes. Advancements in the computing allowed development of new models. Using IBM 650 Multicompartment Dendritic Neuron Model were developed (Wilfrid Rall 1959) (Wilfrid Rall 1962) (Rall et al. 1968). They showed the experimentaly "all or nothing" nature of the spike. Later they produced a series of papers that explained mathematical nature of the action potential.

2.1 Hodgkin-Huxley Model

First, to create a model had to find a suitable candidate and collect data. They focused on the squid giant axon, the big diameter of the axon (0.5mm) allowed them to inject an electrode inside it and measure the voltage difference inside and outside axon (Hodgkin et al. 1939). Later series of papers were published (a, b, c parts) (Hodgkin et al. 1952) that explain the mathematical model in detail. One useful invention "Voltage clamp" allowed Hodgkin and Huxley to conduct productive experiments by controlling voltage inside and outside of the axon membrane. They discovered that when you "Voltage clamp", and fix voltage, there is an inward current at first (Na^+ current), and after some time there will be an outward current (K^+ current). Then they "Voltage clamp" at different levels of voltage, and measured inward and outward currents. Experiments showed that these are voltage-dependent channels. They created an electrical circuit representing a membrane.

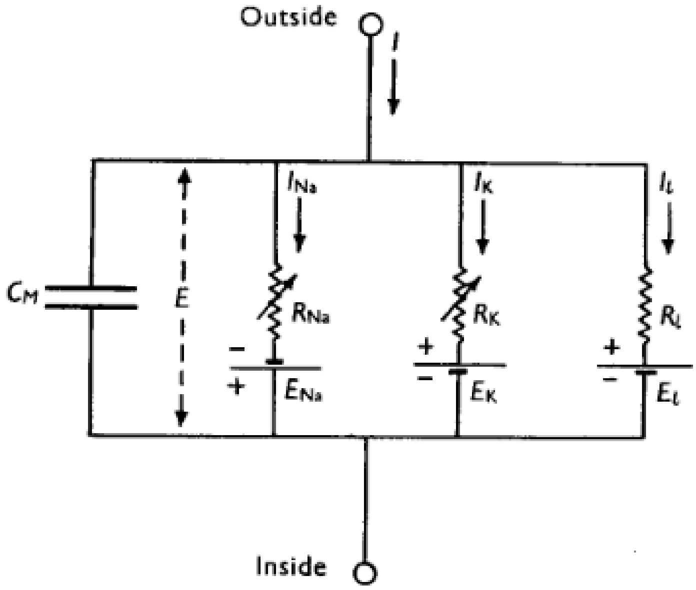

Figure 1. Electronic circuit for Hodkin-Huxley model

In this model membrane is a capacitor. Each channel is represented by a resistor, and extracellular and intracellular liquids are conductors. They formulated an equation based on this diagram:

$$I_m = C\frac{dV}{dt} + g_{Na}(v,t)(V_m - R_N a) + g_{Na}(v,t)(V_m - E_K) + g_L(V_m - E_l).$$

which explains current that flows through the membrane equals the sum of the capacitative current, sodium current, potassium current, and leakage current. Further, they developed equations: for the activation of the sodium:

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

for the deactivation of the sodium:

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

and for activation of potassium:

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

They fitted this equation for experimental data and developed equations for α and β parameters for each equation.

Using this equation they were able to obtain the first simulated action potential.

2.2 Advances Post Hodgkin-Huxley

Since the creation of the Hodgkin-Huxley model new ion channels were revealed, and incorporated into the model which improved accuracy (Hille 1992). New models were developed that incorporated cell types (Nandi et al. 2022).

3 History of Computational Neuron Simulation

One of the earliest attempts to utilize computing in neuron simulation was introduced in GENESIS (Wilson et al. 1989) and NEURON (Hines 1993). Both simulators have GUI and other means to define models. NEURON was developed in C, with the support of HOC for scripting, with later support for Python introduced (Hines 2009). When studies moved from single neurons and small circuits to regions neocortex, simulation became more computationally intensive, and demand for simulators with increased performance arose. The NeoCortical Simulator (NCS) was introduced, which utilizes parallelization. Also, SPLIT simulators were developed (Hammarlund et al. 1998) designed specifically to model large neuron networks utilizing distributed clusters of machines. The increased number of different simulators and complexity created the need for a simulator with a more simple interface, Nest (Gewaltig et al. 2007). A tool for simulating heterogeneous networks of neurons was developed The circuit simulator (CSIM), is now deprecated.

A more recent example of a neuron simulator is Brian2 (Stimberg et al. 2019) which is an easy-to-use simulator with efficient performance.

There were also developments in hardware that allow to simulate of neurons and regions of the brain, so-called "neuromorphic hardware systems". SpiNNaker (Furber et al. 2014) BrainScaleS (Müller et al. 2023). They are inspired by neurons and allow us to run simulations directly on hardware.

Multiple databases were developed to store information about models developed with different software. ModelDB (McDougal et al. 2016), NeuroDB (Zeighami et al. 2021), NeuroMorpho.org (Ascoli et al. 2007).

4 Hands-on Comparison using PyNN

4.1 Introduction to PyNN and its role in neuronal network simulations.

PyNN (Davison 2008) is a simulator independent framework, that allows to describe model once, and run it on different simulators. Currently, only NEURON, NEST, and Brian 2 are supported, and several neuromorphic computers.

Using PyNN we do not need to use specific Python API provided by the specific simulator, and just use PyNN for any simulator(currently only a limited number are supported).

4.2 Creating a Simple Neuron Model

It is easy to create model with PyNN, created model will be translated to the specific name of the targeted simulator.

```
# we import sepcific simulator or a function to get simulator
import pyNN.neuron as sim

#specify initial parameters
sim.setup(timestep=0.01, min_delay=1.0)

#create neurons, specify number of neurons and model
hh_neuron = sim.Population(1, sim.HH_cond_exp())
```

```

#define current, and inject it into neurons
current_source = sim.DCSource(amplitude=0.5, start=20.0, stop=80.0)
hh_neuron.inject(current_source)

#specify what we want to record
hh_neuron.record('v')

#then run simulation, and spicify for how long to run
sim.run(100.0)

#after simulation we can collect information about simulation
data = hh_neuron.get_data().segments[0].analogsignals[0]

#now we can examine data, visualise it, compare different experiments

```

Listing 1. PyNN example

Here is a plot of results:

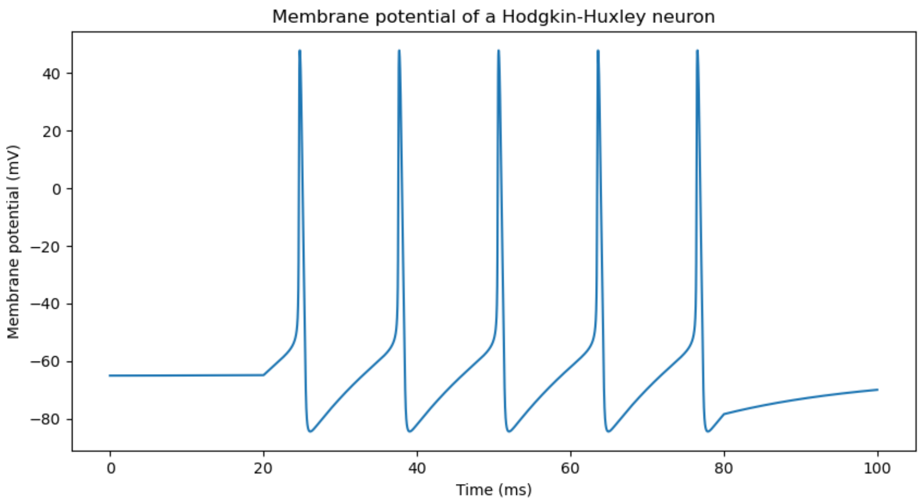

Figure 2. Simulation result using PyNN

4.3 Running Simulations on Different Simulators

A simple simulation of applying current to a single neuron (similar to the Hodgkins-Huxley experiment) is created, and run on NEURON and Brain2 simulator. Running on Nest presented numerous issues.

The first execution took much more time, due to load time and probably Python issues, so only consequent runs were recorded. Execution speed and memory usage were recorded. Additionally, to simple neuron simulation, a set of examples from PyNN documentation were evaluated.

4.4 Comparison and Results

Execution results:

Table 1. Comparison of Neuron Simulators running with PyNN

Simulator	Time	Memory Usage
NEURON	0:00.73	104,728 KB
Brian2	0:03.29	164,108 KB

Result of simulators running natively without PyNN:

Table 2. Comparison of Neuron Simulators natively using original API of each respective simulator

Simulator	Time	Memory Usage
NEURON	0:00.76	66,156 KB
Brian2	0:06.48	246,076 KB

As we can see NEURON performed much better using PyNN, it was much faster and used less memory. Compared to native implementation NEURON performed almost at the same speed, but used less memory. Brain2 simulator performed much worse, it was slower using PyNN and utilized more memory. Implementation with PyNN took less memory than a native implementation for Brian2. It might be however due to imperfect implementation. Brian2 relies heavily on equations and is more complicated than NEURON implementation. NEURON python API and implementation remain of classic Hodgkin-Huxley experiment.

```
      stim = h.IClamp(soma(0.5))

```

Listing 2. Inject current into neuron

we use a "Clamp" to inject current into a neuron.

Initially plan was to compare also Nest simulator, but it proved to be a complicated task. Running Nest with PyNN introduced multiple errors. Building Nest did not improve the situation. And in the end simulation with Nest remained an unachievable goal.

We still were able to compare one of the oldest simulators NEURON and one of the newest ones Brain2. Results show that a more mature simulator still wins. It is faster takes less memory, has better documentation, has cleaner API, and is much easier to implement.

Simulations were run on a computer with a CPU with 4 cores and 8 G memory, on Debian 12. Bash scripts were used to execute Python scripts and for measurements.

All source code is available at https://github.com/jarekrzdbk/NeuronSimulationBenchmark.

5 Conclusion

Simulation of the brain is not a new topic. The possibility of simulating the brain captivated science fiction writers before actual simulators were developed. Simulation of neurons proved to be an important stepping stone in the development of neuroscience. The development of software simulators allowed scientists to reproduce experiments, and test these experiments multiple times with different variables. Simulations allow us to visualize complex networks of neurons. Advances in computing power allowed us to simulate more complex neuron circuits. In some cases, simulations of the whole organism were developed (Szigeti et al. 2014), and simulation of the region of neocortex (Markram et al. 2015). The importance of simulation software was recognized by the scientific community and the Blue Brain Project and Human Brain Project were launched. A wide variety of software tools were developed for simulation, ranging in targeted usage, complexity, and performance. Recently attempts were made to create a unified API with PyNN to create models with simulator independent focus, so code can be written once and run on different simulators. It should make it easier to create simulations. And while some simulators are harder to use like Brain2, benefits in complexity with other simulators like NEURON are not so evident, since writing code for NEURON is not much more difficult than for PyNN.

References

Ascoli, Giorgio A., Duncan E. Donohue, and Maryam Halavi (Aug. 2007). "NeuroMorpho.Org: A Central Resource for Neuronal Morphologies". In: *The Journal of Neuroscience* 27.35, pp. 9247–9251. DOI: 10.1523/jneurosci.2055-07.2007 (page 4).

Bois-Reymond, Emil du (Jan. 1843). "Vorläufiger Abriss einer Untersuchung über den sogenannten Froschstrom und über die elektromotorischen Fische". In: *Annalen der Physik* 134.1, pp. 1–30. DOI: 10.1002/andp.18431340102 (page 2).

Brette, Romain et al. (July 2007). "Simulation of networks of spiking neurons: A review of tools and strategies". In: *Journal of Computational Neuroscience* 23.3, pp. 349–398. DOI: 10.1007/s10827-007-0038-6 (page 1).

Davison, Andrew P (2008). "PyNN: a common interface for neuronal network simulators". In: *Frontiers in Neuroinformatics* 2. DOI: 10.3389/neuro.11.011.2008 (page 4).

FitzHugh, Richard (July 1961). "Impulses and Physiological States in Theoretical Models of Nerve Membrane". In: *Biophysical Journal* 1.6, pp. 445–466. DOI: 10.1016/s0006-3495(61)86902-6 (page 2).

Furber, Steve B. et al. (May 2014). "The SpiNNaker Project". In: *Proceedings of the IEEE* 102.5, pp. 652–665. DOI: 10.1109/jproc.2014.2304638 (page 4).

Gewaltig, Marc-Oliver and Markus Diesmann (2007). "NEST (NEural Simulation Tool)". In: *Scholarpedia* 2.4, p. 1430 (pages 1, 4).

Hammarlund, Per and Örjan Ekeberg (1998). In: *Journal of Computational Neuroscience* 5.4, pp. 443–459. DOI: 10.1023/a:1008893429695 (page 4).

Hille, Bertil (July 1992). "Ionic channels of excitable membranes (second edition)". In: *FEBS Letters* 306.2-3, pp. 277–278. DOI: 10.1016/0014-5793(92)81020-m (page 3).

Hines, Michael (1993). "NEURON — A Program for Simulation of Nerve Equations". In: *Neural Systems: Analysis and Modeling*. Springer US, pp. 127–136. DOI: 10.1007/978-1-4615-3560-7_11 (pages 1, 4).

– (2009). "NEURON and Python". In: *Frontiers in Neuroinformatics* 3. DOI: 10.3389/neuro.11.001.2009 (page 4).

Hodgkin, A. L. and A. F. Huxley (Oct. 1939). "Action Potentials Recorded from Inside a Nerve Fibre". In: *Nature* 144.3651, pp. 710–711. DOI: 10.1038/144710a0 (page 2).

– (Aug. 1952). "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of Physiology* 117.4, pp. 500–544. DOI: 10.1113/jphysiol.1952.sp004764 (pages 1, 2).

Lapicque, LM (1907). "Recherches quantitatives sur l'excitation electrique des nerfs". In: *J Physiol Paris* 9, pp. 620–635 (page 2).

Markram, Henry et al. (Oct. 2015). "Reconstruction and Simulation of Neocortical Microcircuitry". In: *Cell* 163.2, pp. 456–492. DOI: 10.1016/j.cell.2015.09.029 (pages 1, 7).

McDougal, Robert A. et al. (Sept. 2016). "Twenty years of ModelDB and beyond: building essential modeling tools for the future of neuroscience". In: *Journal of Computational Neuroscience* 42.1, pp. 1–10. DOI: 10.1007/s10827-016-0623-7 (page 4).

Müller, Eric et al. (2023). "The BrainScaleS-2 Neuromorphic Platform — A Report on the Integration and Operation of an Open Science Hardware Platform within EBRAINS". DOI: 10.5281/ZENODO.8375521 (page 4).

Nagumo, J., S. Arimoto, and S. Yoshizawa (Oct. 1962). "An Active Pulse Transmission Line Simulating Nerve Axon". In: *Proceedings of the IRE* 50.10, pp. 2061–2070. DOI: 10.1109/jrproc.1962.288235 (page 2).

Nandi, Anirban et al. (Aug. 2022). "Single-neuron models linking electrophysiology, morphology, and transcriptomics across cortical cell types". In: *Cell Reports* 40.6, p. 111176. DOI: 10.1016/j.celrep.2022.111176 (page 3).

Rall, W and G M Shepherd (Nov. 1968). "Theoretical reconstruction of field potentials and dendrodendritic synaptic interactions in olfactory bulb." In: *Journal of Neurophysiology* 31.6, pp. 884–915. DOI: 10.1152/jn.1968.31.6.884 (page 2).

Rall, Wilfrid (Nov. 1959). "Branching dendritic trees and motoneuron membrane resistivity". In: *Experimental Neurology* 1.5, pp. 491–527. DOI: 10.1016/0014-4886(59)90046-9 (page 2).

– (Mar. 1962). "Electrophysiology of a Dendritic Neuron Model". In: *Biophysical Journal* 2.2, pp. 145–167. DOI: 10.1016/s0006-3495(62)86953-7 (page 2).

Stimberg, Marcel, Romain Brette, and Dan FM Goodman (Aug. 2019). "Brian 2, an intuitive and efficient neural simulator". In: *eLife* 8. DOI: 10.7554/elife.47314 (page 4).

Suryawanshi, Chandani R. and Vinod Nayyar (Nov. 2013). "BLUE BRAIN". In: *INTERNATIONAL JOURNAL OF MANAGEMENT &amp INFORMATION TECHNOLOGY* 7.2, pp. 1009–1017. DOI: 10.24297/ijmit.v7i2.3294 (page 1).

Szigeti, BalÃ¡zs et al. (Nov. 2014). "OpenWorm: an open-science approach to modeling Caenorhabditis elegans". In: *Frontiers in Computational Neuroscience* 8. DOI: 10.3389/fncom.2014.00137 (pages 1, 7).

Wilson, MA and JM Bower (1989). "The simulation of large-scale neural networks Methods in Neuronal Modeling ed C Koch and I Segev" (pages 1, 4).

Zeighami, Sepanta and Cyrus Shahabi (2021). "NeuroDB: A Neural Network Framework for Answering Range Aggregate Queries and Beyond". DOI: 10.48550/ARXIV.2107.04922 (page 4).