Bhadra-Kilgore Block of Dorsal Root Ganglion Neurons to Address Chronic Pain

Christian Leonard

BIOMEDE 517- Neural Engineering

April 17th, 2022

Introduction

According to the CDC, 20.4% of US citizens suffer from chronic pain every day, making it one of the most widespread health conditions in the United States. Of those who suffer from chronic pain, 25% will develop chronic pain syndrome. People with CPS have drastically increased rates of anxiety, depression, and suicide. The United States alone reports the economic impact of chronic pain – including paying for medical treatments, lost wages, loss of productivity, etc. – is $635 Billion, annually.

The current means of combating chronic pain are limited and problematic. 70% of all abused prescription drugs are painkillers, such as oxycodone, hydrocodone, or fentanyl. Of these painkiller abusers, 65.7% report that they take the painkillers as intended [1]. Addiction has become a growing byproduct of pain treatment. Painkillers have also contributed heavily to the current opioid crisis in the United States, as 8 out of 10 heroin users say their addiction started after abusing prescription painkillers.

Dorsal root ganglion stimulation is a new means of treating chronic pain that does not require the use of opioid painkillers, but instead, opts for electrical stimulation of neurons in the dorsal root ganglion (DRG) of the lower spinal cord. According to the neurosurgery department at PennState Health, roughly 75% of patients that received the new DRG stimulation treatment reported “significant relief” of their symptoms. Thus far, the DRG stimulation procedure has focused on those whose pain originates within the lower body (foot, knee, hip, groin, and lower abdomen) [2]. A more thorough examination of the neurons within the DRG, as well as the sort of stimulation that is able to hinder the reception of chronic pain, could not only open doors to expand this treatment, but could be the key to helping reduce the global weight of chronic pain, drug abuse, and the negative economic affiliated associated with this condition.

Methods

Previous models of the DRG neuron were implemented in a variety of programming languages, with the most noteworthy being NEURON. The programming language of choice for this simulation is NEURON utilized through Google CoLab. The main challenge in recreating previous models of DRG neurons comes in the form of an incomplete port of NEURON into Python through Google CoLab. Key functions, such as Frankenhaeuser-Huxley channels that were so heavily relied upon in previous models of the DRG found on ModelDB, are not present in Google CoLab’s version of NEURON. This resulted in a shift from Frankenhaeuser-Huxley to Hodgkin-Huxley for all neurons modeled in this project.

The main motivation for the DRG neuron simulated in this project is the model published by Amir R, Devor M (2003) [3]. The main takeaways from this were the ion values of the soma and axon [4]. The inner and outer values for the sodium and potassium ions in the model helped increase the accuracy of the DRG simulation used in this project.

The general structure of the DRG neuron was similar to that of the Mainen neuron simulated in Lab 2 of this course. The key difference between the Mainen neuron and the DRG neuron is twofold. First, the DRG model has a single axon instead of one part myelinated, one part non-myelinated as is the case in the Mainen neuron. Second, the axon and the dendrites of the DRG neuron are also considerably longer and larger in diameter compared to the Mainen neuron. The soma of the DRG neuron is also quite small in comparison to the other parts of the DRG neuron.

Readings were taken at the beginning, directly in the middle, and at the tail end of the axon. This was done to monitor the success of any neuron electrical block employed upon the DRG neuron. Being able to differentiate the readings from the different portions of the axon within the neuron allows for a better understanding of the impact that a neuron block has on the DRG neuron's ability to reach an action potential given a stimulus.

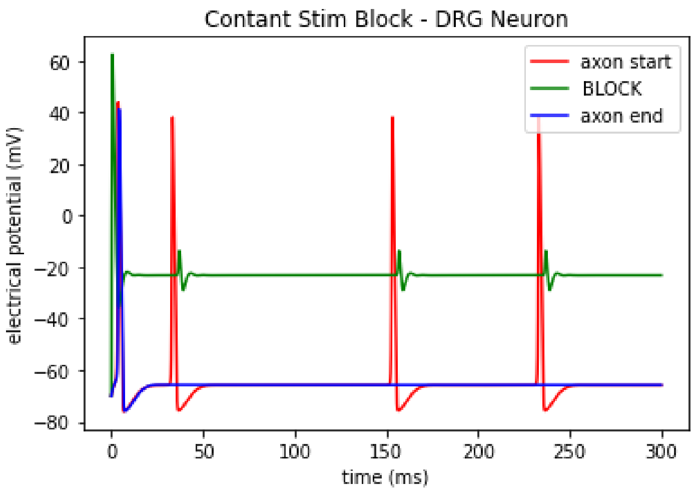
The Bhadra-Kilgore block is a means of neurostimulation that is capable of blocking action potentials from propagating completely throughout a neuron by essentially cutting off the electrical current at the block site. This occurs because the ion channels of the neuron are kept in a state where they are unable to open or close as needed to cause the neuron to fire under electrical stimulation. During testing, the Bhadra-Kilgore nerve block was shown to be effective on 100% of the tested neurons and is fully reversible, making it the ideal means of blocking the DRG's action potential [5].

A simplified, first iteration of a depolarizing block was first performed as a foundation to prove the block's ability to perform on a DRG neuron. Though the depolarizing block described in Bhadra and Kilgore (2004) is an oscillating sine wave current with a frequency of around 3 kHz, this first version was just a constant signal that utilized NEURON's 'IClamp' functionality [3]. The complexity of the block was then increased to better resemble a true Bhadra-Kilgore block. The goal of each block was simple: stop the action potential of neurons, thereby reducing chronic pain signals being processed through the DRG in the spinal cord.

Results

Though the DRG's underlying structure borrows heavily from the Mainen neuron, they are quite different. For instance, the DRG neuron saw an action potential occur at an amp value of 1080, whereas the Mainen neuron required only 7.7 for it to reach an action potential. With an action potential occurring and the segments of the DRG neuron being differentiable on a graph, the first iteration of the Bhadra-Kilgore block was then implemented to try and stop the action potential from taking place.

Is a depolarizing block able to be successfully applied to a DRG neuron? That was the first major question of this project and a key factor in determining how DRG stimulation treatment is able to be successful. To proceed, a simplified 'IClamp' block ran for the full duration of the simulation, constantly outputting a flat amperage value to the middle of the axon. The 'IClamp' block was successful at stopping the neuron from reaching an action potential on the DRG neuron. The following figure shows the 'IClamp' block working as intended on the DRG neuron model.



This graph shows the change in electrical potential that the DRG neuron experiences over time, measured in millivolts and milliseconds, respectively. The changes in potential occur as a result of stimulation currents applied to the neuron in attempts to cause an action potential, represented as a spike on the graph.

As described by the legend, the red line signifies the reading at the start of the axon. This is the part of the neuron that receives the action potential stimulation before the block is reached. As such, it makes sense that the first part of the axon would see enough stimulation to initiate a spike. From this graph, we can see that four of these initial stimulation tests were applied to the neuron throughout the 300 milliseconds the neuron was tested.

The green line represents the blocking current applied to the neuron by the ‘IClamp’ block. This line brings the resting potential of the neuron from its starting potential of -70 mV to just under -20 mV. This potential is elevated enough that the neuron is unable to fully propagate the stimulation signal. This can be seen in the small wave in the green line that occurs after each red spike on the graph – the first portion of the axon’s response to each trial’s stimulation in an effort of reaching an action potential. While the red spike reaches heights of 40 mV, the block – once active – doesn’t even break into positive millivolts.

The blue line is representative of the voltage reading at the tail of the axon. The blue line has two distinct behaviors on this graph. The first is the spike that occurs within the first 8 ms of the simulation. **Figure 2** shows these first seconds in greater detail. This occurs due to the block (the green line) having an initialization period. In figure 2, the block doesn’t reach its resting potential until after the neuron has reached its action potential. In the case of chronic pain mitigation, this would, unfortunately, result in pain still being felt upon activation of the block as the line does, indeed, spike. The green line in figure 2 also shows up well before the red or blue lines because the block has no delay added to it, while the first test stimulation has a brief delay.

The second phase of the blue line is the steady resting at its default potential of -70 mV. The fact that the blue line doesn’t react to any of the red spikes indicates a functional block, as the trial stimulations are unable to cross the block and reach the tail end of the axon. **Figure 3** shows the behavior of the DRG neuron with all 4 trial stimulations present if the 'IClamp' block is turned off. In this instance, each stimulation of the neuron results in a full action potential occurring, and by extension, pain signals being received within the DRG. Here, the red, green, and blue lines all propagate in order (start, middle, and tail-end of the axon), confirming the action potential's propagation through the entire neuron.

After the 'IClamp' proved successful at blocking action potentials, the complexity of the block was increased to resemble the block described in (Bhadra, Kilgore 2004). The Bhadra-Kilgore block performed similarly to that of the 'IClamp' block when tested using the same DRG neuron model under the same four trial pulses. **Figure 4** shows the Bhadra-Kilgore block's performance during these trials. It is incredibly similar to figure 1 shown above, with all colored lines maintaining their same meanings. The noticeable difference comes as a result of the red spike. In figure 4, the green line has slightly more movement when blocking the red spike's signal than in figure 1's green line. The green line also shows more of its oscillating qualities when first activating during the first 20 ms of the simulation, displaying more movement in the line when compared to the 'IClamp' block's graph. **Figure 5** shows that the first few milliseconds of the Bhadra-Kilgore block are nearly identical to that of the 'IClamp' block.

Is the performance of the Bhadra-Kilgore favorable to a constant stimulation block such as the 'IClamp' block? The results of this simulation suggest yes, the Bhadra-Kilgore block is favorable from an energy usage standpoint. Though the Bhadra-Kilgore opts for the oscillating sine waveform over the 'IClamp' block, the amplitude required for the Bhadra-Kilgore block to function was lower than that of the 'IClamp' block. With implants having small batteries, a higher amperage value that is consistently on would require a greater energy expenditure than a lower amp oscillation block, like the Bhadra-Kilgore, increasing the cost and complexity of the implant greatly.

Discussion

Though the results of this paper are quite promising for a functional neurostimulation block on a DRG neuron, it is still quite a simplified version of the DRG neuron that is used. The lack of support for Google CoLab's port of NEURON in Python, such as not having support for Frankenhaeuser-Huxley or .HOC files – both of which were heavily used in the DRG models found on ModelDB – results in a questionably accurate DRG neuron representation for the block to be applied. These gaps were made up as best as possible by utilizing Hodgkin-Huxley channels or by simplifying the neuron to not have to rely on the standard import .HOC files. Given more time, the detail of the DRG neuron should be prioritized by either switching the simulation over to the full version of NEURON or by performing more tests to verify the neuron's accuracy.

Should the neuron prove accurate and the block function as intended, it is a strong first step toward creating a chronic pain treatment that doesn't utilize potentially addictive painkillers, is more effective at blocking the pain, and doesn't impair users. Globally, treating pain in this manner could improve the quality of life for so many, reduce suicide rates, and drastically reduce the number of new drug abusers that stem from the abuse of painkillers.

References

1. “Prescription Drug Abuse Statistics.” NCDAS, 6 Apr. 2022, https://drugabusestatistics.org/prescription-drug-abuse-statistics/.

2. Huygen, F. J., Kallewaard, J. W., Nijhuis, H., Liem, L., Vesper, J., Fahey, M. E., ... & Capobianco, R. A. (2020). Effectiveness and safety of dorsal root ganglion stimulation for the treatment of chronic pain: a pooled analysis. Neuromodulation: Technology at the Neural Interface, 23(2), 213-221.

3. Amir R, Devor M (2003) Electrical excitability of the soma of sensory neurons is required for spike invasion of the soma, but not for through-conduction. Biophys J 84:2181-91

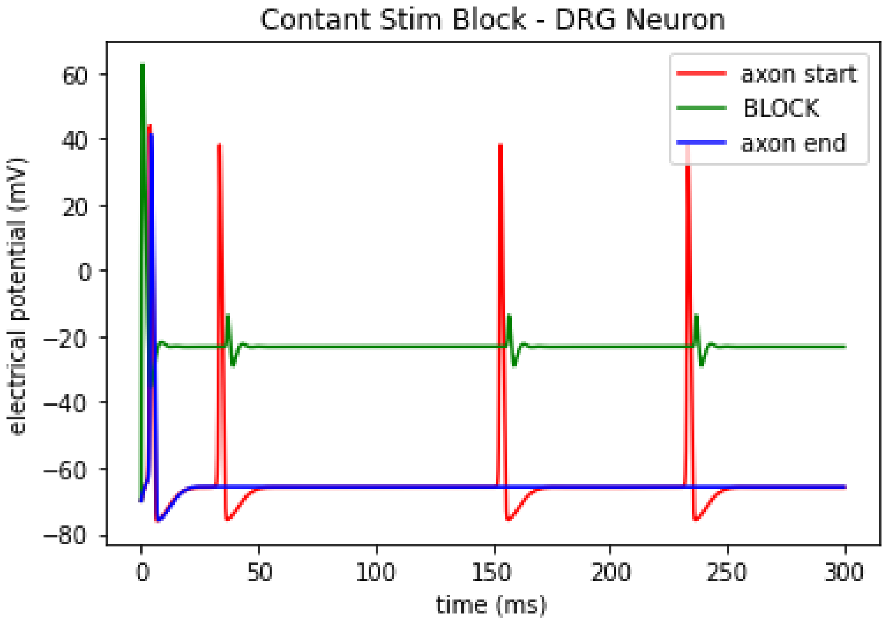
4. Source code is available in ModelDB (Amir, Devor 2003) at https://senselab.med.yale.edu/ModelDB/ShowModel?model=51022&file=/amirdevor03/CellNOhill.hoc#tabs-2

5. Bhadra, N., & Kilgore, K. L. (2004). Direct current electrical conduction block of peripheral nerve. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 12(3), 313-324

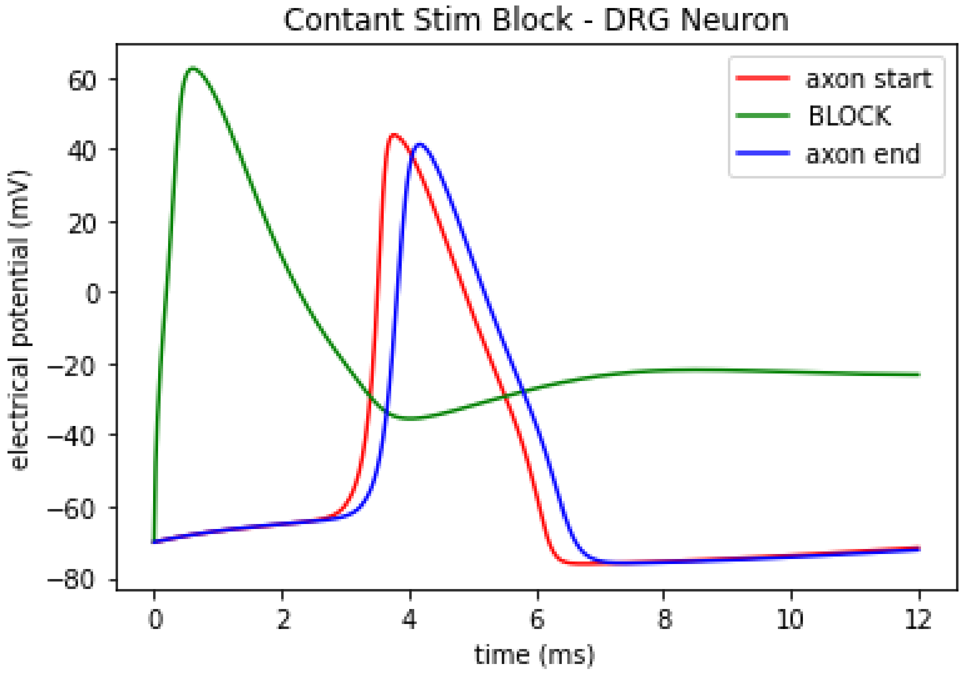
Appendix

A. FIGURES

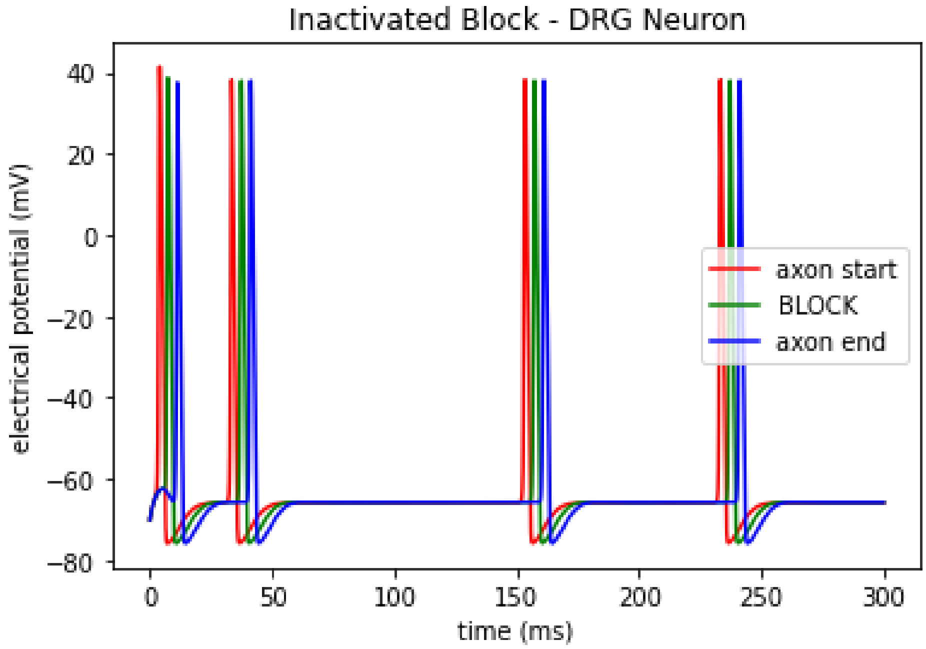
1. Functional IClamp stimulus block on the DRG neuron model



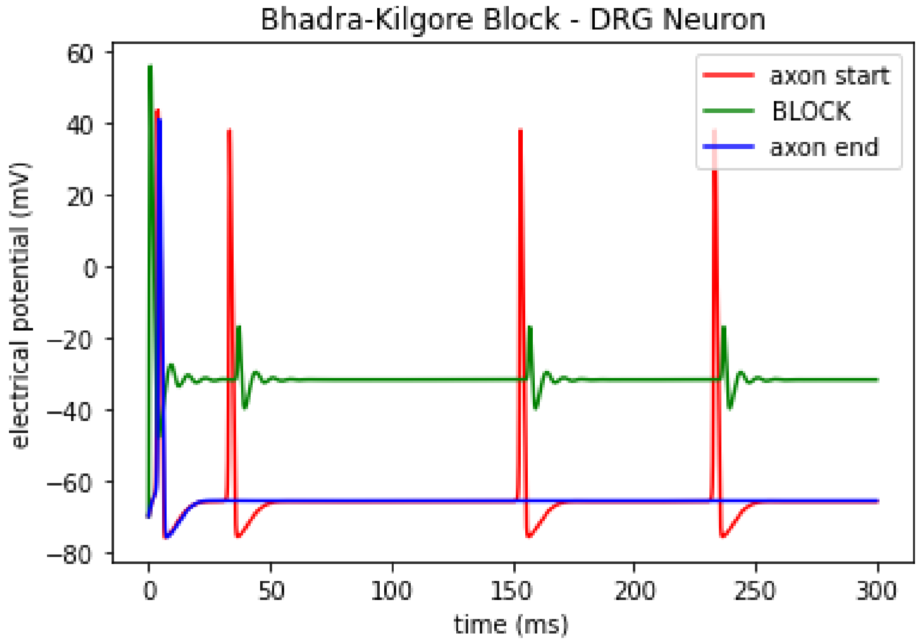
2. Initialization of IClamp block on DRG neuron model



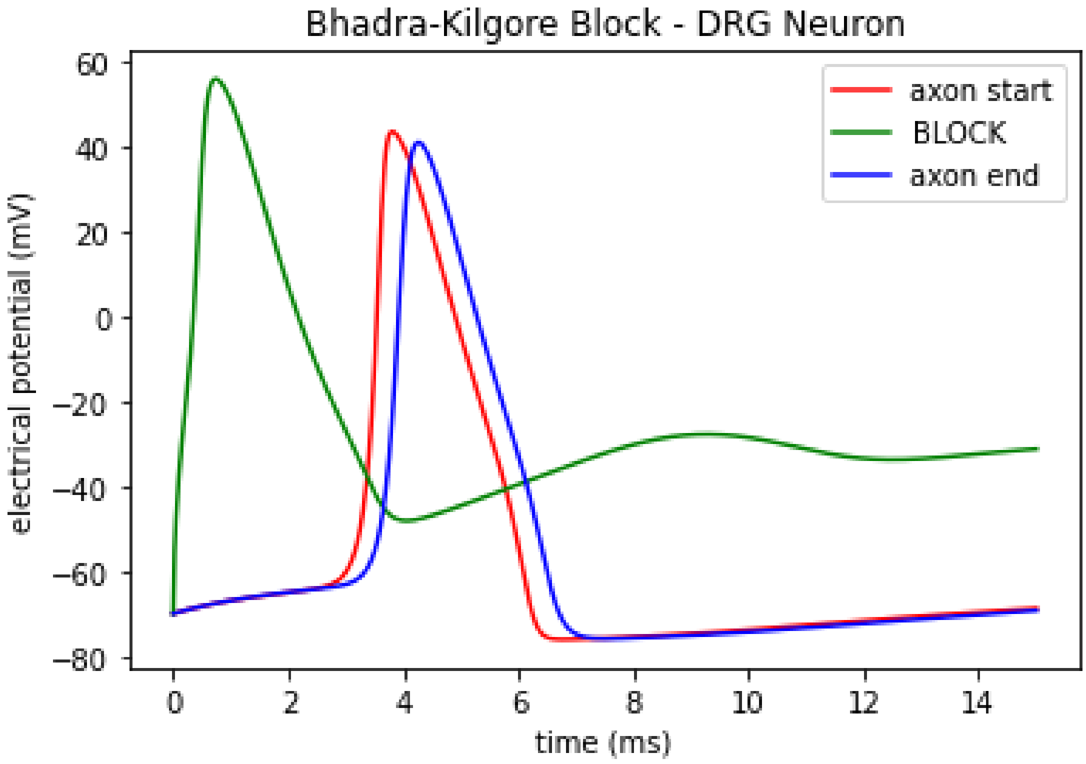
3. Inactive ‘IClamp’ block on DRG neuron model



4. Functional Bhadra-Kilgore Block on DRG neuron model



5. Initialization of Bhadra-Kilgore block on DRG neuron model



B. CODE

1. **Simplified Axon and 'IClamp' stimulation block**

```
"""517 final - simple

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1WClNZYf7FMzc4-7u5mi4UN33D5yc-x_d
"""

!pip install neuron
from neuron import h, gui
from neuron.units import ms, mV, um
from math import e #Euler's number
import math
import matplotlib.pyplot as plt
import numpy as np
import scipy.io
h.load_file("stdrun.hoc")

axon = h.Section(name = 'axon')
axon.nseg = 100
axon.L = 10000 * um
axon.diam = 20 * um
axon.insert(h.hh)
axon.insert(h.pas)
axon.insert(h.extracellular)
axon.Ra = 200 #ohm-cm**2

axon.g_pas = 0.0002
axon.nai = 13.74
axon.nao = 114.5
axon.ki = 120
axon.ko = 2.5

"""-- BLOCK --
- [DONE] Make an IClamp that runs the entire duration of the simulation
- [DONE] The neuron should sit at -70 mV before the block with no
stimulus, -40 mV with the block on, and -70 mV after the block
- [DONE] Plot the middle of the axon when the block is engaged
"""

#block - sine wave
#block = h.IClamp(axon(0.5))
#block.delay = 0 * ms
#block.dur = 1e9 * ms

## per Bhadra, Kilgore 2004: The most efficient waveform for conduction
block was a 3-5 kHz constant-current biphasic sinusoid

#freq = 3 #KHz
#steps = 1/freq #period
#dt = steps/50 #num points in sine wave

#t_block = h.Vector(np.linspace(0, steps, 50))
#y_block = h.Vector((71.2/2) * (np.sin((2* np.pi / steps) * t_block)))
#y_block.play(axon(0.5)._ref_ina, t_block, True)

#for i in range(0, len(y_block)):
  #block.amp = np.asarray(y_block[i])

block = h.IClamp(axon(0.5))
block.delay = 4 * ms
block.dur = 1e3 * ms
block.amp = 65

stim = h.IClamp(axon(0.01))
stim.delay = 0 * ms
stim.dur = 0.1 * ms
stim.amp = 50

trial1 = h.IClamp(axon(0.01))
trial1.delay = 20 * ms
trial1.dur = 0.1 * ms
trial1.amp = 50

trial2 = h.IClamp(axon(0.01))
```

```
trial2.delay = 120 * ms
trial2.dur = 0.1 * ms
trial2.amp = 50

trial3 = h.IClamp(axon(0.01))
trial3.delay = 250 * ms
trial3.dur = 0.1 * ms
trial3.amp = 50

v1 = h.Vector().record(axon(0.1)._ref_v)
t1 = h.Vector().record(h._ref_t)

v2 = h.Vector().record(axon(0.5)._ref_v)
t2 = h.Vector().record(h._ref_t)

v3 = h.Vector().record(axon(0.9)._ref_v)
t3 = h.Vector().record(h._ref_t)

h.finitialize(-70 * mV) #resting membrane potential
h.continuerun(300 * ms)

plt.plot(t1, v1, 'r', label = 'axon start')
plt.plot(t2, v2, 'g', label = 'BLOCK')
plt.plot(t3, v3, 'b', label = 'axon end')

plt.xlabel('time (ms)')
plt.ylabel('membrane voltage (mV)')
plt.title('Contant Block Stim - Simplified Axon')
plt.legend()
plt.show()
```

2. DRG Neuron Model with Functional ‘IClamp’ Block

```
Original file is located at

https://colab.research.google.com/drive/1xIqwwS0W1X7K7dlr60Bx2rrk-B8A9f2r
"""

!pip install neuron
from neuron import h
from neuron.units import ms, mV, um
from math import e #Euler's number
import math
import matplotlib.pyplot as plt
import numpy as np
import scipy.io
h.load_file("stdrun.hoc")

"""---DRG NEURON FORM---
- Needs an axon hillock?
- Once the block works on the standard, parietal neuron above, then
transfer the stimulus injections to the DRG neuron below to see if the
block is effective
"""

dend = h.Section(name = 'dend')
dend.L = 5000 * um
dend.diam = 80 * um
dend.nseg = 400
dend.insert(h.hh)
dend.insert(h.pas)
dend.insert(h.extracellular)

soma = h.Section(name = "soma")
soma.L = 20 * um
soma.diam = 80 * um
soma.nseg = 1 #ModelDB has 1 seg
soma.insert(h.hh)
soma.insert(h.pas)
soma.insert(h.extracellular)
soma.nai = 13.74
soma.nao = 114.5
soma.ki = 120
soma.ko = 2.5
soma.gl_hh = 0.0002

hillock = h.Section(name = 'hillock')
hillock.L = 16 * um
hillock.nseg = 9

i = 80
```

```
j = 1
while j < 10:
  hillock(j * 0.1).diam = i * um
  j = j + 1 #all 9 segments
  i = i - 7.5 #even spacing from soma to axon (80 - 5 um)

hillock.insert(h.hh)
hillock.insert(h.pas)
hillock.insert(h.extracellular)

axon = h.Section(name = 'axon')
axon.L = 10000 * um
axon.diam = 20 * um
axon.nseg = 400 # 4,000 segments -- ModelDB (2003) is only *6* segments
axon.insert(h.hh)
axon.insert(h.pas)
axon.insert(h.extracellular)
axon.Ra = 200 #ohm-cm**2
axon.g_pas = 0.0002
axon.nai = 13.74
axon.nao = 114.5
axon.ki = 120
axon.ko = 2.5

soma.connect(dend)
hillock.connect(soma)
axon.connect(hillock)

block = h.IClamp(axon(0.5))
block.delay = 0 * ms
block.dur = 1e3 * ms
block.amp = 90 #90 is the min for blocking to work on this current neuron

stim = h.IClamp(dend(0))
stim.delay = 1 * ms
stim.dur = 0.1 * ms
stim.amp = 1080

trial1 = h.IClamp(dend(0))
trial1.delay = 30 * ms
```

```
trial1.dur = 0.1 * ms
trial1.amp = 1080

trial2 = h.IClamp(dend(0))
trial2.delay = 150 * ms
trial2.dur = 0.1 * ms
trial2.amp = 1080

trial3 = h.IClamp(dend(0))
trial3.delay = 230 * ms
trial3.dur = 0.1 * ms
trial3.amp = 1080

v1 = h.Vector().record(axon(0.1)._ref_v)
t1 = h.Vector().record(h._ref_t)

v2 = h.Vector().record(axon(0.5)._ref_v)
t2 = h.Vector().record(h._ref_t)

v3 = h.Vector().record(axon(0.9)._ref_v)
t3 = h.Vector().record(h._ref_t)

h.finitialize(-70 * mV)
h.continuerun(300 * ms)

plt.plot(t1, v1, 'r', label= 'axon start')
plt.plot(t2, v2, 'g', label= 'BLOCK')
plt.plot(t3, v3, 'b', label= 'axon end')

plt.xlabel('time (ms)')
plt.ylabel('electrical potential (mV)')
plt.title('IClamp Block - DRG Neuron')
plt.legend()
plt.show()
```

3. DRG Neuron Model with Functional Bhadra-Kilgore Block

```
Original file is located at


https://colab.research.google.com/drive/1iNauwFlzUVS1J1c0Fq81UO4OB1W3L59j
"""
```

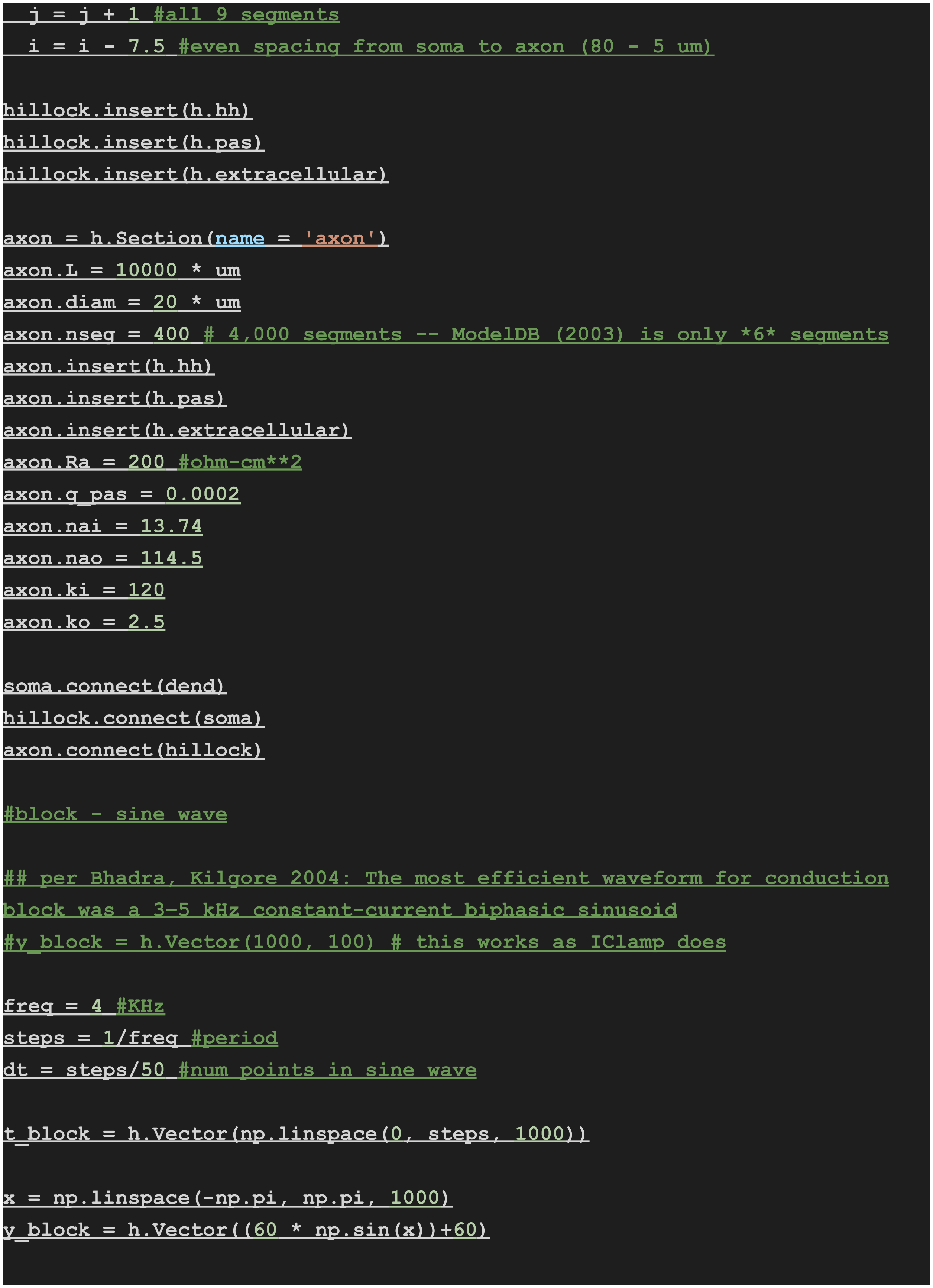
```
!pip install neuron
from neuron import h
from neuron.units import ms, mV, um
from math import e #Euler's number
import math
import matplotlib.pyplot as plt
import numpy as np
import scipy.io
h.load_file("stdrun.hoc")

dend = h.Section(name = 'dend')
dend.L = 5000 * um
dend.diam = 80 * um
dend.nseg = 400
dend.insert(h.hh)
dend.insert(h.pas)
dend.insert(h.extracellular)

soma = h.Section(name = "soma")
soma.L = 20 * um
soma.diam = 80 * um
soma.nseg = 1 #ModelDB has 1 seg
soma.insert(h.hh)
soma.insert(h.pas)
soma.insert(h.extracellular)
soma.nai = 13.74
soma.nao = 114.5
soma.ki = 120
soma.ko = 2.5
soma.gl_hh = 0.0002

hillock = h.Section(name = 'hillock')
hillock.L = 16 * um
hillock.nseg = 9

i = 80
j = 1
while j < 10:
  hillock(j * 0.1).diam = i * um
```



```
  j = j + 1 #all 9 segments
  i = i - 7.5 #even spacing from soma to axon (80 - 5 um)

hillock.insert(h.hh)
hillock.insert(h.pas)
hillock.insert(h.extracellular)

axon = h.Section(name = 'axon')
axon.L = 10000 * um
axon.diam = 20 * um
axon.nseg = 400 # 4,000 segments -- ModelDB (2003) is only *6* segments
axon.insert(h.hh)
axon.insert(h.pas)
axon.insert(h.extracellular)
axon.Ra = 200 #ohm-cm**2
axon.g_pas = 0.0002
axon.nai = 13.74
axon.nao = 114.5
axon.ki = 120
axon.ko = 2.5

soma.connect(dend)
hillock.connect(soma)
axon.connect(hillock)

#block - sine wave

## per Bhadra, Kilgore 2004: The most efficient waveform for conduction
block was a 3–5 kHz constant-current biphasic sinusoid
#y_block = h.Vector(1000, 100) # this works as IClamp does

freq = 4 #KHz
steps = 1/freq #period
dt = steps/50 #num points in sine wave

t_block = h.Vector(np.linspace(0, steps, 1000))

x = np.linspace(-np.pi, np.pi, 1000)
y_block = h.Vector((60 * np.sin(x))+60)
```

```
y_block.play(axon(0.5)._ref_ina, t_block, True)

block = h.IClamp(axon(0.5))
block.delay = 0 * ms
block.dur = 1e3 * ms

for i in range(0, len(y_block)):
  block.amp = (y_block[i])

plt.plot(t_block, y_block)

stim = h.IClamp(dend(0))
stim.delay = 1 * ms
stim.dur = 0.1 * ms
stim.amp = 1080

trial1 = h.IClamp(dend(0))
trial1.delay = 30 * ms
trial1.dur = 0.1 * ms
trial1.amp = 1080

trial2 = h.IClamp(dend(0))
trial2.delay = 150 * ms
trial2.dur = 0.1 * ms
trial2.amp = 1080

trial3 = h.IClamp(dend(0))
trial3.delay = 230 * ms
trial3.dur = 0.1 * ms
trial3.amp = 1080

v1 = h.Vector().record(axon(0.1)._ref_v)
t1 = h.Vector().record(h._ref_t)

v2 = h.Vector().record(axon(0.5)._ref_v)
t2 = h.Vector().record(h._ref_t)

v3 = h.Vector().record(axon(0.9)._ref_v)
t3 = h.Vector().record(h._ref_t)
```

```
h.finitialize(-70 * mV)
h.continuerun(15 * ms)

plt.plot(t1, v1, 'r', label= 'axon start')
plt.plot(t2, v2, 'g', label= 'BLOCK')
plt.plot(t3, v3, 'b', label= 'axon end')

plt.xlabel('time (ms)')
plt.ylabel('electrical potential (mV)')
plt.title('Bhadra-Kilgore Block - DRG Neuron')
plt.legend()
plt.show()
```