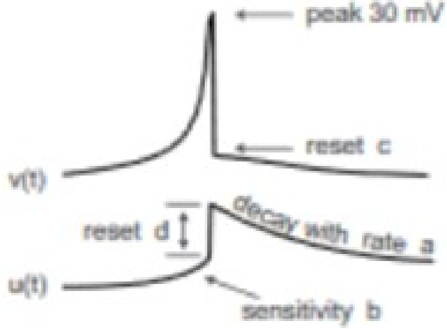
Simulating the Izhikevich spiking neuron model using the Brian2 software

Work by: Alex Beltran and Jorge Orbegozo.
Presentation by: Alex Beltran.

Description of the problem

Main task: Use Brian2 to model the Izhikevich spiking neuron patterns.

Figure 1: Izhikevich's spiking neuron model.



Description of the problem

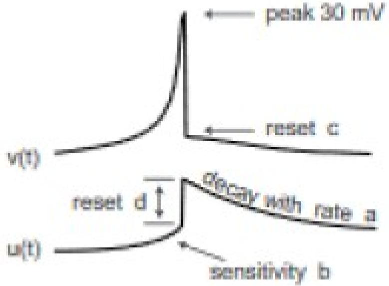
Izhikevich model:

- The Izhikevich model is an specific model for spiking neurons.
- Uses a two-dimensional system of ordinary differential equations.
- As a reset, a second system defines for when v reaches its threshold value (30).

$$v' = 0.04v^2 + 5v + 140 - u + I$$
$$u' = a(bv - u)$$
$$v' = \frac{dv}{dt}\ , u' = \frac{du}{dt}$$

$$v = c$$
$$u = u + d$$

Figure 1: Izhikevich's spiking neuron model.



Description of the problem

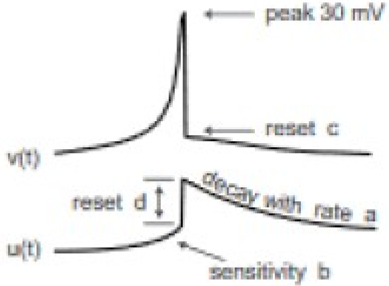
Izhikevich model:

- **The parameters:**
  - V is membrane potential, U is membrane recovery.
  - A is the time scale of U, b relates the sensitivity of U to V.
  - C is the reset value of V, D is the effect spiking has on U on reset.

$$v' = 0.04v^2 + 5v + 140 - u + I$$
$$u' = a(bv - u)$$
$$v' = \frac{dv}{dt}\,,\; u' = \frac{du}{dt}$$

$$v = c$$
$$u = u + d$$

Figure 1: Izhikevich's spiking neuron model.



Description of our approach:

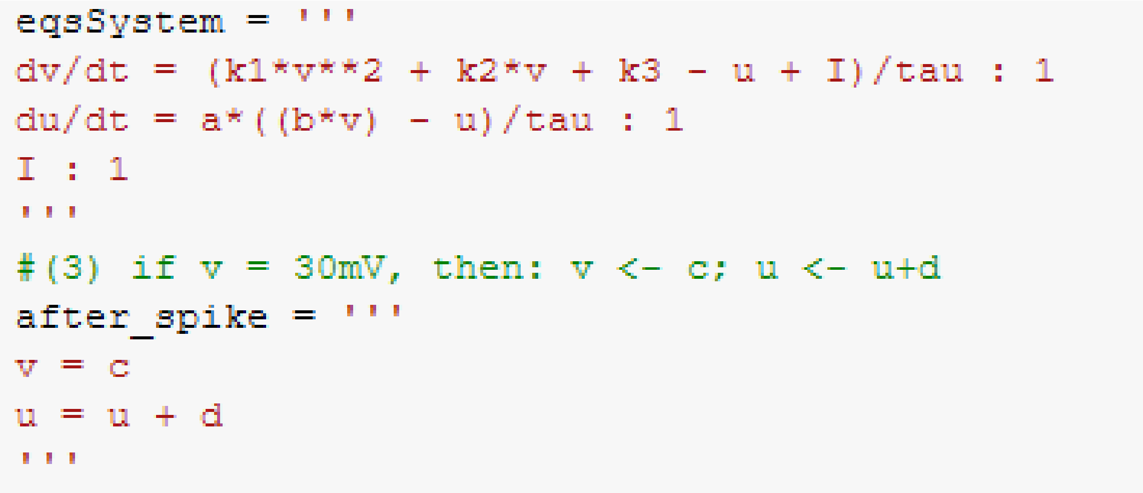
The equation in our model:

- **Problems:**
  - The “constants” are not constant.
  - Brian2 requires a time step.

$$
\begin{gathered}
v' = 0.04v^2 + 5v + 140 - u + I \\
u' = a(bv - u) \\
v' = \frac{dv}{dt} \text{ , } u' = \frac{du}{dt}
\end{gathered}
\longrightarrow
\begin{gathered}
v' = (k1v^2 + k2v + k3 - u + I)/tau \\
u' = a((bv) - u)/tau \\
v' = \frac{dv}{vt} \text{ , } u' = \frac{du}{dt}
\end{gathered}
$$

Description of our approach:

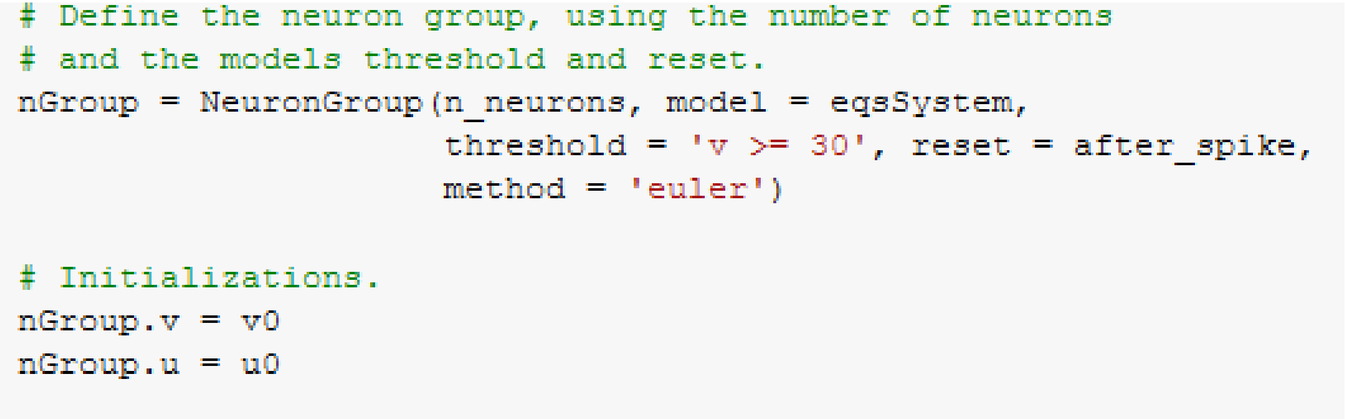
In brian2:



```
eqsSystem = '''
dv/dt = (k1*v**2 + k2*v + k3 - u + I)/tau : 1
du/dt = a*((b*v) - u)/tau : 1
I : 1
'''
#(3) if v = 30mV, then: v <- c; u <- u+d
after_spike = '''
v = c
u = u + d
'''
```

Description of our approach:

In brian2:

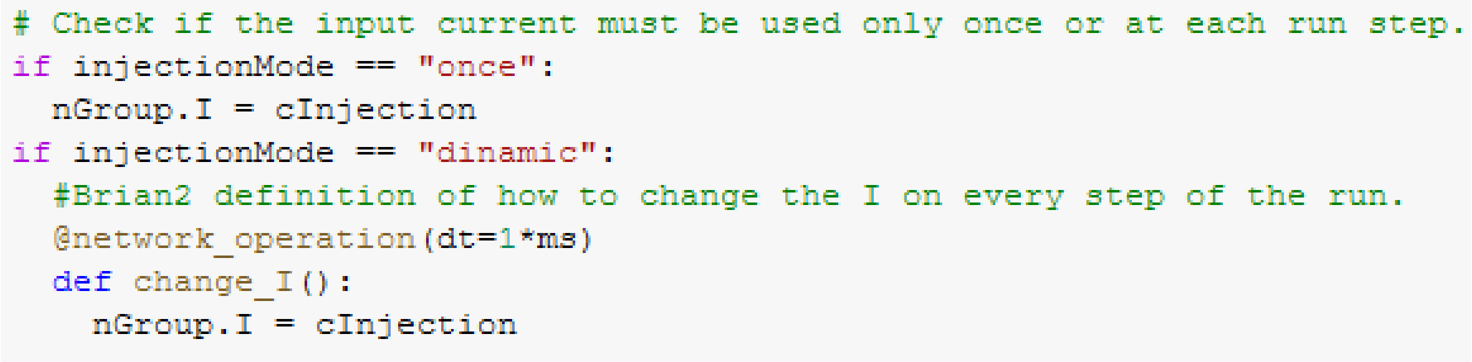


```
# Define the neuron group, using the number of neurons
# and the models threshold and reset.
nGroup = NeuronGroup(n_neurons, model = eqsSystem,
                     threshold = 'v >= 30', reset = after_spike,
                     method = 'euler')

# Initializations.
nGroup.v = v0
nGroup.u = u0
```

Description of our approach:

In brian2:



```
# Check if the input current must be used only once or at each run step.
if injectionMode == "once":
  nGroup.I = cInjection
if injectionMode == "dinamic":
  #Brian2 definition of how to change the I on every step of the run.
  @network_operation(dt=1*ms)
  def change_I():
    nGroup.I = cInjection
```

Results:

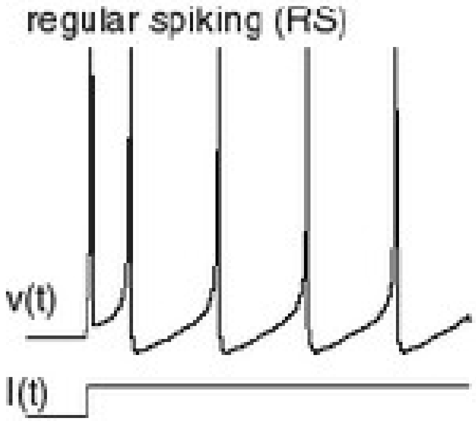


Figure 2: Izhikevich original model result.

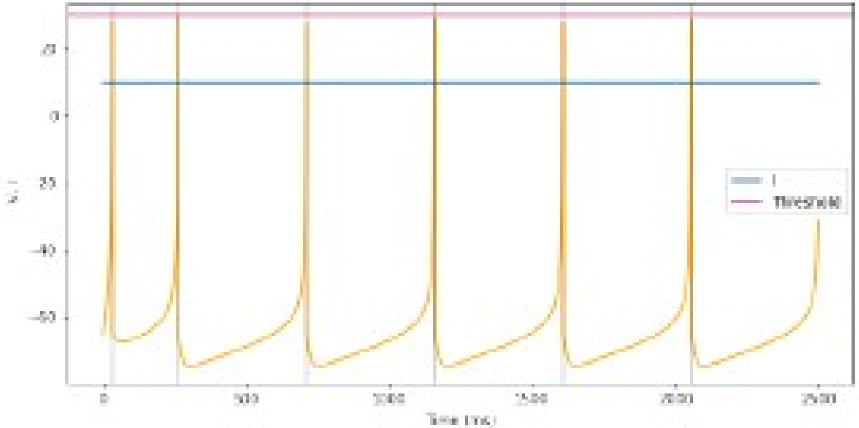


Figure 3: Our model's result.

Parameters used: Default configuration and $d = 8$.

Results:

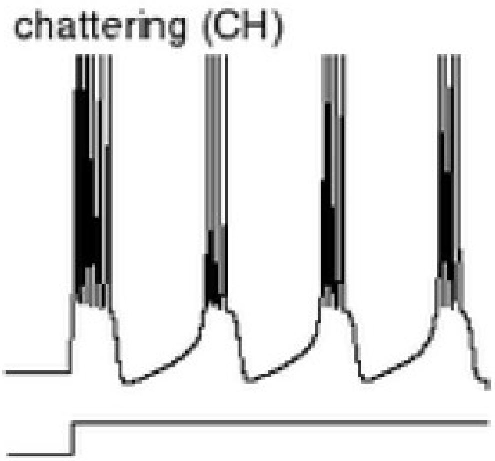


Figure 6: Izhikevich original model result.

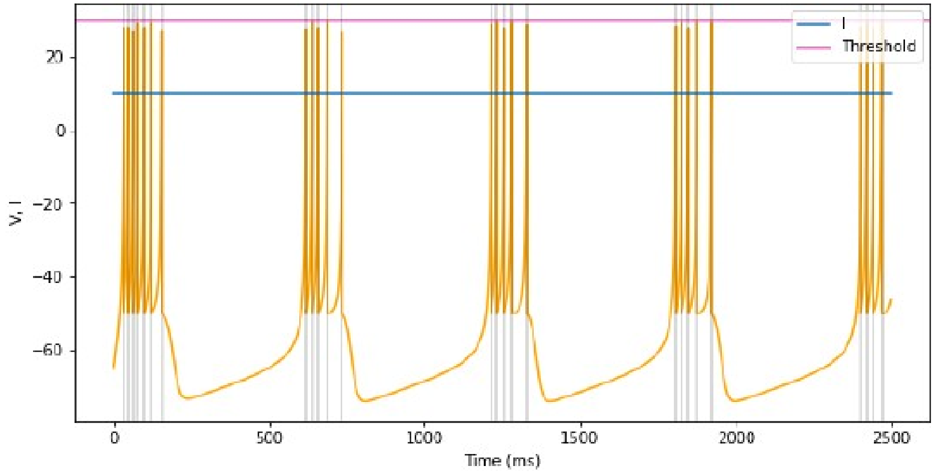


Figure 7: Our model's result.

Parameters used: Default configuration and $c = -50$.

Results:

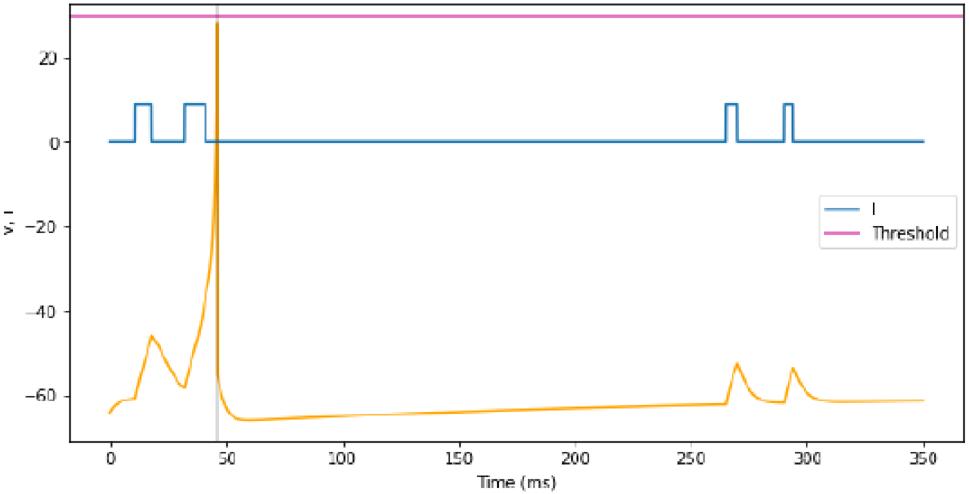


Figure 27: (L) Integrator.

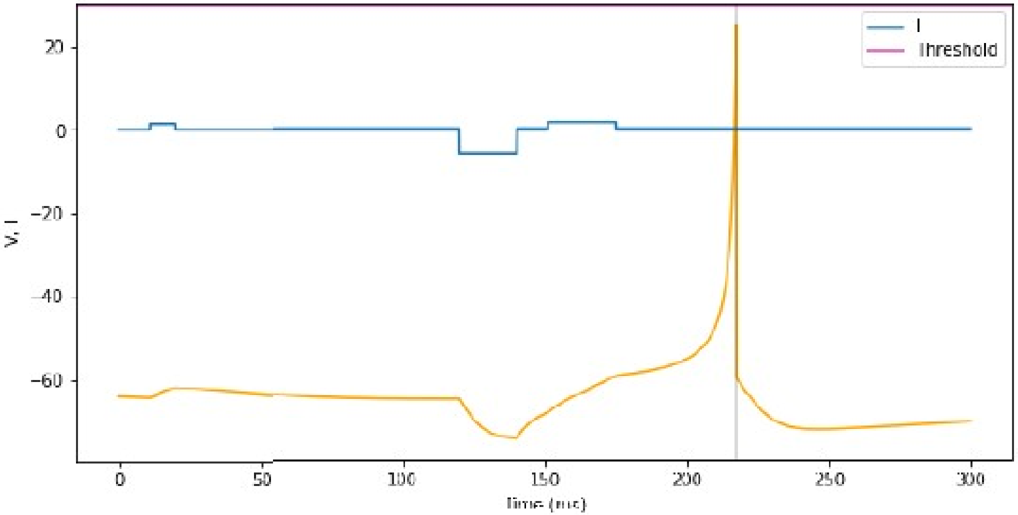


Figure 30: (O) Threshold variability.

Conclusions:

1. **From the 26 patterns, we could correctly replicate every single one but 2 of them.**

   1. (R) Accommodation and (RZ) Resonator.

2. **Mixed feelings on the Izhikevich model.**

   1. The good: extremely simple to implement and really fast model that could replicate lots and lots of neuron patterns.

   2. The bad: The abuse of what we call “magic numbers”.

3. **About brian2:**

   1. Powerful and reliable.

   2. Really good error back-tracing system.