THE HODGKIN-HUXLEY MODEL AS A PERFORMANCE METRIC

Joshua Vega

November 8, 2018

In the world of scientific computing, languages like Python, Julia, and even MATLAB have become the de facto standard. It's understandable why these languages have become so popular too. For one, you don't have to be a computer scientist to use them well; just about anyone can pick them up and start building useful tools. In addition, they're all interpreted languages and don't need to be compiled before being executed making it faster to iterate on.

While these languages have a lot going for them does performance suffer? Performance may not always be the most important metric when selecting a language to use, but as models become more complex performance starts to matter. In this post, I'll be implementing the well-known Hodgkin-Huxley model in several different programming languages to see how their performance compares.

The Hodkin-Huxley model is a mathematical model that describes the propagation of action potentials in neurons. It was developed by Alan Lloyd Hodgkin and Andrew Fielding Huxley in 1952 to describe the giant axon in squid. Since then, it has become a fairly standard model in mathematical neuroscience. I won't be discussing the model itself in much more detail since we're really only concerned with it's performance but below are a few of the important equations. I do however highly recommend the Wikipedia page for a brief but thorough discussion on it.

$$C_m \frac{dV_m}{dt} = I - I_K - I_{Na} - I_L$$

$$I_K = g_K n^4 (V_m - V_K)$$

$$I_{Na} = g_{Na} m^3 h (V_m - V_{Na})$$

$$I_L = g_L (V_m - V_L)$$