

The FitzHugh-Nagumo model

To recap, we have seen the Hodgkin Huxley equation

$$C_m \frac{dV}{dt} = g_\ell(E_\ell - V) + g_{\text{Na}}(E_{\text{Na}} - V) + g_{\text{K}}(E_{\text{K}} - V) + I \quad (1)$$

where I is an external current. The sodium and potassium conductances, g_{Na} and g_{K} have complicated non-linear dynamics. In fact, the Hodgkin-Huxley equation is really four equations, an equation for V along with equations for the three gating variables n , m and h , each of the form

$$\frac{dk}{dt} = \alpha_k(1 - k) - \beta_k k \quad (2)$$

with k standing in for n , m or h and $\alpha_k(V)$ and $\beta_k(V)$ being the probability of going from closed to open and open to closed. By moving stuff around this can be easily rewritten in a familiar form

$$\tau_k \frac{dk}{dt} = k_\infty - k \quad (3)$$

with

$$\tau_k = \frac{1}{\alpha_k + \beta_k} \quad (4)$$

and

$$k_\infty = \frac{\alpha_k}{\alpha_k + \beta_k} \quad (5)$$

Hence the ion channels relax towards some asymptotic value n_∞ , m_∞ and h_∞ with some time scale τ_n , τ_m and τ_h ; however all these quantities depend on the voltage so the equations are coupled to the voltage equation.

This is a complicated non-linear equation; in fact, real neurons are likely to be even more complicated. As was mentioned before; the Hodgkin-Huxley equation as described here describes the voltage dynamics of the squid giant axon. The same name is often used to describe the more complex versions of these models

$$C_m \frac{dV}{dt} = \text{lots of different ionic currents} \quad (6)$$

used to describe real neurons. These typically include other channels, often more than one potassium channel which adds other timescales to the dynamics, calcium channels which play some of the same role as the sodium

channels, other sodium channels with more complex dynamics, calcium-gated potassium channels whose open and closing depends not on the voltage but on the concentration of calcium ions, and so on. All of this complexity emphasises the neuron itself as a site of computation, rather than as a node in a network. As such, it is important to understand the dynamics of the neurons models, however, their complexity makes this hard and so we start not by allowing the Hodgkin-Huxley model to become more complex, but by simplifying it.

FitzHugh-Nagumo model

The FitzHugh-Nagumo model [1, 2] is a set of two differential equations that show similar behaviour to the Hodgkin-Huxley equations: they are held to capture some of the dynamical structure of the Hodgkin-Huxley equations, something that can be explored using phase analysis. Here, however, we are going to restrict ourselves to solving the equations numerically and observing the spiking.

The FitzHugh-Nagumo model is governed by the equations

$$\begin{aligned}\frac{dv}{dt} &= v - \frac{1}{3}v^3 - w + I \\ \tau \frac{dw}{dt} &= v + a - bw\end{aligned}\tag{7}$$

where a little- v , v , has been used for the voltage to show these quantities shouldn't be taken seriously as biologically relevant, they have been scaled to, for example, get rid of one of the time constants. The little- w , w , should be thought of as modelling the net effect of the other currents in the Hodgkin-Huxley equations.

One application of this is the study of bursting cells and pattern generation; these are cells that send out regular bursts of neurons; these dynamics are important in controlling some fundamental physiological systems where patterns are important, chewing in slugs, struggling in tadpoles and so on.

In this coursework we will solve the equation numerically with the parameters $I = 0.5$, $a = 0.7$, $b = 0.8$, and $\tau = 12.5$. This is a bit different from the previous example of the Euler method because now we have two variables to update, v and w , corresponding to v and w . This isn't too complicated, each gets updated using the derivative in the usual way.

One complication is that they get updated ‘at the same time’, in other words, you should do something like:

```
v_old = v
w_old = w
```

and then

```
v += v_derivative(v_old, w_old) * dt
w += w_derivative(v_old, w_old) * dt
```

so that both variables get updated using the values from the previous time step. If you just did

```
v += v_derivative(v, w) * dt
w += w_derivative(v, w) * dt
```

then v would be updated using the old values, whereas w would be updated using the new value of v !

References

- [1] FitzHugh R. (1955) Mathematical models of threshold phenomena in the nerve membrane. Bull. Math. Biophysics, 17:257–278
- [2] Nagumo J., Arimoto S., and Yoshizawa S. (1962) An active pulse transmission line simulating nerve axon. Proc. IRE. 50:2061–2070.