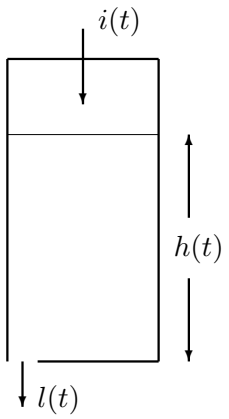
Introduction

These notes are about first order ordinary differential equations. These equations appear frequently in neuroscience so it is useful to consider them now before going any further.

Buckets of water

In the simplest model of neurons their voltage dynamics is similar to the dynamics of a bucket with a leak and the class of equations that apply in this case will also be applied to synapses, for example.



Consider a bucket with straight sides which is filled to a height h with water. Imagine the water leaks out of a hole in the bottom. The rate the water leaks out depends on h; the larger h is the larger the pressure at the bottom is and hence the faster the water pours out. In other words

$$l(t) \propto h(t) \tag{1}$$

or

$$l(t) = Gh(t) \tag{2}$$

where G is a constant which will depend on the size of the hole and complicated things like the viscosity of water. Of course, we are also ignore lots of complicated stuff, like turbulence and so forth, but since we are interested in the equation rather than the amount of water in a bucket, this is fine. Imagine water also pours in the top at a rate $i(t)$. This means the total rate of change of the amount of water is $i(t) - Gh(t)$.

Now, $h(t)$ is the height of the water not the volume: the volume is $Ch(t)$ where C is the cross-sectional area of the bucket. The rate of change of the volume is therefore

$$\frac{dCh(t)}{dt} = i(t) - Gh(t) \tag{3}$$

or

$$\frac{dh}{dt} = \frac{1}{C}(i - Gh) \tag{4}$$

Let's solve this equation for constant i before going on to look at neurons. Probably the best to do this is by using an integrating factor, let $\tau = C/G$ and $\tilde{\imath} = i/G$

$$\tau \frac{dh}{dt} + h = \tilde{\imath} \tag{5}$$

then we multiply across by $\exp t/\tau$ and divide by τ

$$e^{t/\tau}\frac{dh}{dt} + \frac{1}{\tau}e^{t/\tau}h = \frac{\tilde{\imath}e^{t/\tau}}{\tau} \tag{6}$$

Now we can rewrite the left hand side using the product rule

$$\frac{d}{dx}(uv) = u\frac{dv}{dx} + v\frac{du}{dx} \tag{7}$$

to give

$$\frac{d}{dt}\left(e^{t/\tau}h\right) = \frac{\tilde{\imath}e^{t/\tau}}{\tau} \tag{8}$$

Now integrating both sides gives

$$e^{t/\tau}h = \tilde{\imath}e^{t/\tau} + A \tag{9}$$

where A is an integration constant. This gives

$$h = Ae^{-t/\tau} + \tilde{\imath} \tag{10}$$

and putting $t = 0$ shows $A = h(0) - \tilde{\imath}$ so

$$h(t) = [h(0) - \tilde{\imath}]e^{-t/\tau} + \tilde{\imath} \tag{11}$$

so, basically, the value of h decays exponentially until it equilibriates with $\tilde{\imath}$.

These dynamics make good intuitive sense; the more water there is in the bucket, the higher the pressure will be at the leak and the quicker the water will pour out. If there is just the right amount of water the rate the water pours out the leak will precisely match the rate it pours in, this is the equilibrium. If there is more water than required for equilibrium it will pour out faster than the flow coming in, if there is less, it will pour out slower. Either way, as time passes the height of the water will reach the equilibrium. The plot in Fig. 1 illustrates this.

Variable input

We have only discussed constant inputs; the variable input case where i depends on time is harder and although it can sometimes be solved it is often easier just to compute it numerically, we will look briefly at how to do this, but first note that the effect of variable input is that the solution kind of chases the input with a timescale set by τ. For very small τ is chases it quickly, so it is close to the input, but for large τ it lags behind it and smooths it out. This is sometimes described by saying that it *filters* the input. There is an illustration in Fig. 2.

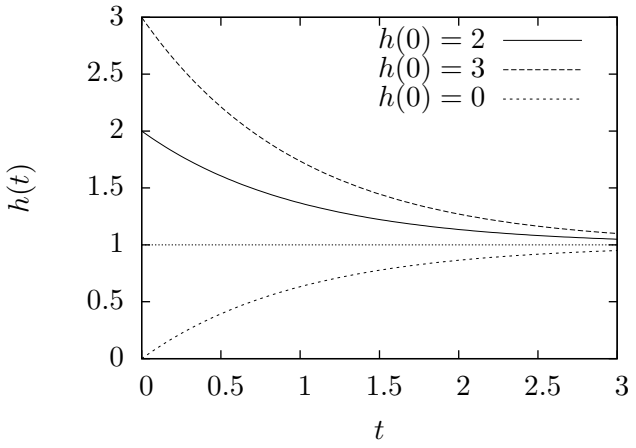


Figure 1: Exponential relaxation. The dynamics described by the 'bucket equation' are very common. Here $h = [h(0) - \tilde{\imath}]\exp(-t/\tau) + \tilde{\imath}$ is plotted with $\tilde{\imath} = 1$, $\tau = 1$ and three different values of $h(0)$. $h(t)$ relaxes towards the equilibrium value $\tilde{\imath} = 1$, the closer it gets, the slower it approaches.

Numerical solutions

Consider the problem of solving a general first order differential equation

$$\frac{df(t)}{dt} = F(f,t) \tag{12}$$

This includes our bucket case if we replace f with h and $F(f,t)$ with $(i - Gh)/C$. Now consider the situation where we want to solve this but can't do it analytically, as will be the case for many values of the time dependent input $i(t)$ or, indeed, if $i(t)$ itself is only known numerically.

The simplest standard numerical approach is Euler's method. Say you know the value of $f(t)$ and want to approximate $f(t + \delta t)$ where δt is some small increment in time. Now, we know from the Taylor expansion that

$$f(t + \delta t) = f(t) + \delta t \frac{df(t)}{dt} + \frac{1}{2}\delta t^2 \frac{d^2 f(t)}{dt^2} + O(\delta t^3) \tag{13}$$

Now we know what $df(t)/dt$ is, it is $F(f,t)$ from the equation, so

$$f(t + \delta t) = f(t) + \delta t F(f,t) + O(\delta t^2) \tag{14}$$

so approximating $f(t+\delta t)$ with $f(t)+\delta t F(f,t)$ is actual to first order in the time step δt. This is the Euler approximation. More formally if we have an initial condition $f(t_0)$ and write

$$f_n = f(t_0 + n\delta t) \tag{15}$$

$$t_n = t_0 + n\delta t \tag{16}$$

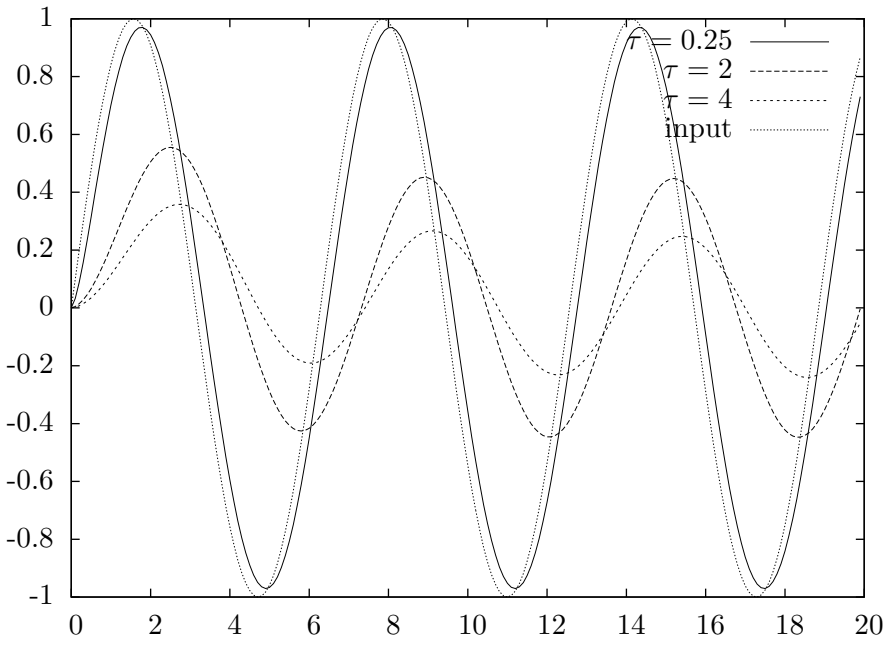


Figure 2: Variable input. Here the input is a sine wave $\tilde{\imath} = \sin t$ and the equation is evolved with $h(0)$ and three different τ values. For $\tau = 0.25$ we see $h(t)$ closely matches the input whereas for larger τ it is smoother and lags behind.

then the Euler approximation is

$$f_{n+1} = f_n + \delta t F(f_n, t_n) \tag{17}$$

Thus, in the case we are interested in here

$$h_{n+1} = \frac{[i(t_n) - Gh_n]\delta t}{C} \tag{18}$$

Now this approximation is often good enough for neuroscience, especially if as often happens the errors are sometimes positive and sometimes negative. However, it can also happen that they can add up and the $O(\delta t^2)$ error can be a problem. The way around this is to use Runge-Kutta. Roughly, for Runge-Kutta a number of different Taylor expansions are considered, using not just the expansion with δt but also $\delta t/2$ and combining them in such a way as to get an approximation that is accurate to a higher order in δt. In other words it combines different quantities so that the coefficients of δt^2 and so on cancel. The Euler method can be thought of as a first order Runge Kutta method, the method that is used most often is fourth order Runge Kutta, the error for this is $O(\delta t^5)$. Explicitly, see for example [], first define k_1 through to k_4:

$$k_1 = \delta t F(f_n, t_n) \tag{19}$$
$$k_2 = \delta t F(f_n + k_1/2, t_n + \delta t/2) \tag{20}$$
$$k_3 = \delta t F(f_n + k_2/2, t_n + \delta t/2) \tag{21}$$
$$k_4 = \delta t F(f_n + k_3, t_n + \delta t) \tag{22}$$

and then

$$f_(n+1) = f_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \tag{23}$$

In fact, in neuroscience, we sometimes have to deal with what are called *stiff* differential equations; these are equations where change is sometimes very rapid and sometimes much slower. Obviously the bucket equations discussed for far don't have this property, in fact they only have one timescale, τ, but later we will example the Hodgkin-Huxley equations, which do. We will not consider the problem of stiff equations in the course, but there are methods, such as the *backwards Euler* method that deal well with stiff equations and the standard neuron simulation packages, like NEURON and GENESIS, and BRIAN with Python, are optimized to the particular properties of the differential equations governing neuronal dynamics.

References

[1] Press WH, Teukolsky SA, Vetterling WT and Flannery BP. (1988) "Numerical recipes in C." Cambridge University Press.