Numerical integration of differential equations

This chapter is about the Taylor series and about the numerical solution of differential equations; two methods are considered, the Euler method and one of the Runge Kutta methods. These are the most straight-forward way to solve differential equations on a computer, there are other methods that are commonly used in neuroscience, like backwards Runge Kutta and adaptive time step methods, but these are easy enough to understand in the future once you know about Euler and Runge Kutta.

The Taylor series

As an introduction it is often useful to use a series description of functions. This can help with numerical calculations of the values and it can be useful in studying properties of the function. Here, it will be used to work out how to efficiently calculate the solutions to differential equations. The Taylor series is one commonly applicable approach to representing a function as a series.

Imagine you have a function $f(t)$ that can be represented as a series

$$f(t) = \sum_{n=0}^{\infty} a_n t^n = a_0 + a_1 t + a_2 t^2 + \dots \tag{1}$$

Now, putting $t = 0$ we get

$$f(0) = a_0 \tag{2}$$

Next, differentiate

$$\frac{df(t)}{dt} = a_1 + 2a_2 t + 3a_3 t^2 + \dots = \sum_{n=1}^{\infty} a_n n t^{n-1} \tag{3}$$

so, putting $t = 0$ we get

$$\frac{df}{dt}|_{t=0} = a_1 \tag{4}$$

Differentiating again gives

$$\frac{d^2 f(t)}{dt^2} = 2a_2 + 6a_3 t + 12a_4 t^2 \dots = \sum_{n=2}^{\infty} a_n n(n-1) t^{n-2} \tag{5}$$

so

$$\frac{1}{2}\frac{d^2 f}{dt^2}|_{t=0} = a_2 \tag{6}$$

and so on.

In fact, by this sort of calculation we see that

$$a_n = \frac{1}{n!}\left.\frac{d^n f}{dt^n}\right|_{t=0} \tag{7}$$

or, put another way,

$$f(t) = \sum_{n=0}^{\infty} \frac{1}{n!}\left.\frac{d^n f}{dt^n}\right|_{t=0} t^n \tag{8}$$

This is the Taylor series. We haven't proven that $f(t)$ has a series of the form $\sum_{n=0}^{\infty} a_n t^n$ at all and not all functions do, in particular, if the function is badly behaved at $t = 0$ it may not. We also haven't proved that the series converges. If we write

$$f(t) = \sum_{n=0}^{N-1} \frac{1}{n!}\left.\frac{d^n f}{dt^n}\right|_{t=0} t^n + E_N(t) \tag{9}$$

where $E_N(t)$ represents the error from stopping at after N terms, we might expect $E_N(t)$ vanishes as N goes to infinity. In fact, this doesn't always happen and sometimes, even when the series does converge, it does so very slowly, so $E_N(t)$ remains large even for very large values of N. Frequently the series converges for some values of t but not for others. Nonetheless, the Taylor series is often useful.

You might already know the series expansion of $\exp t$, but lets calculate it as a Taylor series. Since

$$\frac{d}{dt}\exp t = \exp t \tag{10}$$

we know

$$\frac{d^n}{dt^n}\exp t = \exp t \tag{11}$$

or

$$\left.\frac{d^n}{dt^n}\exp t\right|_{t=0} = 1 \tag{12}$$

for all n so

$$f(t) = \sum_{n=0}^{\infty} \frac{1}{n!} t^n \tag{13}$$

Next let's consider

$$f(t) = \sin t \tag{14}$$

Now

$$\frac{d}{dt} f(t) = \cos t \tag{15}$$

and

$$\frac{d^2}{dt^2} f(t) = -\sin t \tag{16}$$

and so on. Putting $t = 0$ and using $\sin 0 = 0$ and $\cos 0 = 1$ we get

$$\sin t = \sum_{n\,\text{odd}} \frac{(-1)^{(n-1)/2} t^n}{n!} \tag{17}$$

Finally, we have been expanding around $t = 0$, but you can expand around any point, here we expand around $t = t_0$

$$f(t) = \sum_{n=0}^{\infty} \frac{1}{n!} \left.\frac{d^n f}{dt^n}\right|_{t=t_0} (t - t_0)^n \tag{18}$$

or, writing $\epsilon = t - t_0$

$$f(t_0 + \epsilon) = \sum_{n=0}^{\infty} \frac{1}{n!} \left.\frac{d^n f}{dt^n}\right|_{t=t_0} \epsilon^n \tag{19}$$

Numerical solutions of differential equations

Consider the differential equation

$$\frac{df}{dt} = G(t, f) \tag{20}$$

This class of differential equations would include the equation we looked at before:

$$\frac{df}{dt} = -\frac{1}{\tau} f \tag{21}$$

with $G(f) = -f/\tau$ or

$$\frac{df}{dt} = \frac{1}{\tau}[g(t) - f] \tag{22}$$

with $G(t, f) = (g - f)/\tau$. In fact, for most of this discussion we will restrict ourselves to the case where G doesn't depend on t except through f, this case is a small bit simpler.

Now imagine we want to find numerical values for $f(t)$ where we know $f(0) = f_0$, some value, and

$$\frac{df}{dt} = G(f) \tag{23}$$

Imagine further that we can't solve the equation analytically, so we resort to solving it approximately on the computer. This might be necessary because the equation is too hard to solve, or, in the case of the equation

$$\tau \frac{df}{dt} = g(t) - f(t) \tag{24}$$

it might be that we don't know $g(t)$ analytically.

The normal approach would be to discretize time and to work out the solution approximately for each time step in turn, depending on the previous time step. In other words, say we choose δt, a small value, as the time step then we would work out $f(\delta t)$, then use that to work out $f(2\delta t)$ and so on. Now, if δt is small, then δt^2 will be smaller and δt^3 smaller still; the idea behind numerical solutions to differential equations is that we drop higher powers of δt.

Let us use the notation $f_n = f(n\delta t)$ and consider how we might get a computer to work out f_{n+1} approximately if f_n is already known. Now, by the Taylor expansion

$$f(n\delta t + \delta t) = f(n\delta t) + \left.\frac{df}{dt}\right|_{t=n\delta t} \delta t + \frac{1}{2} \left.\frac{d^2 f}{dt^2}\right|_{t=n\delta t} (\delta t)^2 + \dots \tag{25}$$

so one simple approach is to ignore the $(\delta t)^2$ and smaller terms, since $df/dt = G(f)$ this gives

$$f_{n+1} = f_n + G(f_n)\delta t \tag{26}$$

This approximation is known as the *Euler method*. A simple example is plotted in Fig. 1.

Since the Euler method takes into account the δt part of the Taylor approximation but not the δt^2 term, we say it is accurate up to $O(\delta t^2)$, in other words

$$f(t + \delta t) = (\text{Euler approximation}) + O(\delta t^2) \tag{27}$$

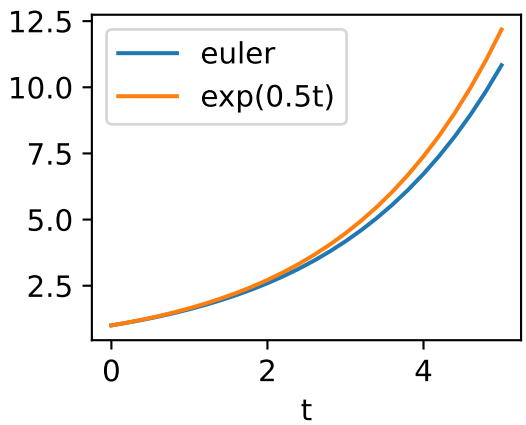

Figure 1: A comparison of the Euler method with the true solution for the equation $df/dt = rf$ with $r = 0.5$. This equation in actually one where the Euler method works quite well for modest values of t, though the error always has the same sign and starts to accumulate, here a large time step of $\delta t = 0.2$ is used to emphasise the error, the actual solution is plotted for comparison.

where, roughly, the $O(\delta t^2)$ stands for stuff that behaves like δt^2 as δt gets small.

The Runge-Kutta method

The Runge-Kutta method uses the Taylor expansion in a clever way to find a better approximation than the Euler method. It is a bit convoluted, so there is a lot of notation, but it does give a very useful numerical algorithm. We are not going to explicitely derive the Runge-Kutta method here; you can see a longer discussion of where it comes from in `runge_kutta.pdf` in the same notes. The key idea behind Runge-Kutta, as with Euler, is to find a way to approximate $f(t+\delta t)$, the Runge-Kutta approximation, however, accounts for more terms of the Taylor expansion. In fact, the fourth order Runge-Kutta we will describe here gets everything up to the fourth order, the errors are like δt^5:

$$f(t+\delta t) = (\text{Runge-Kutta approximation}) + O(\delta t^5) \tag{28}$$

So in the fourth order Runge Kutta approximation we have, as before

$$\frac{df}{dt} = G(f) \tag{29}$$

and we calculate

$$\begin{aligned} k_1 &= G(f_n) \\ k_2 &= G\left(f_n + \frac{1}{2}\delta t k_1\right) \\ k_3 &= G\left(f_n + \frac{1}{2}\delta t k_2\right) \\ k_4 &= G\left(f_n + \delta t k_3\right) \end{aligned} \tag{30}$$

then the approximation is

$$f_{n+1} = f_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\delta t \tag{31}$$

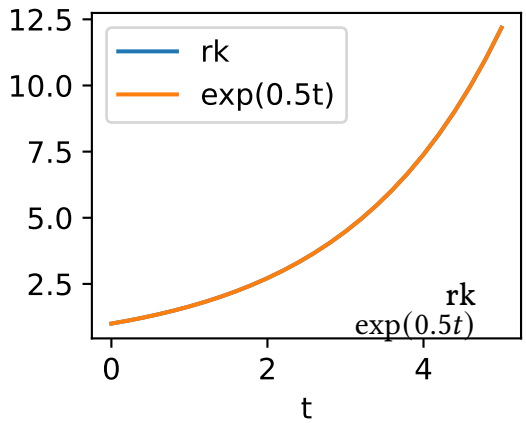

Figure 2: A comparison of the fourth order Runge Kutta method with the true solution for the growth equation. This has the same values of r and δt as in Fig. 1, but the fourth order Runge-Kutta method is used instead of the Euler method. As you can see, the approximation and true solution are indistinguishable.