University of Waterloo MATH 228 notes (second half)

Russell Milne

Autumn 2021

Table of Contents

Chapter 1

Laplace transforms

1.1 November 1: Using Laplace transforms to solve ODEs

Before I begin, I would like to show one more example of taking the Laplace transform. Suppose you have a piecewise continuous function, or in other words a function that is continuous everywhere except for a finite number of jump discontinuities. Remember that the Laplace transform consists of an improper integral, with the upper bound of the integral being infinity:

$$L[f(t)] = F(s) = \int_0^\infty e^{-st} f(t)\ dt \tag{1.1}$$

If you have a piecewise function, it can be written as a bunch of different "pieces":

$$f(t) = \begin{cases} f_1(t),\ 0 \leq t < t_1 \\ f_2(t),\ t_1 \leq t < t_2 \\ \dots \\ f_n(t),\ t_{n-1} \leq t < \infty \end{cases} \tag{1.2}$$

Each of these different pieces can be considered to be zero everywhere outside its particular interval (i.e. $t < t_{n-1}$ and $t \geq t_n$). If a function is zero over some interval, then integrating it over that interval will produce a result of zero. Hence, the Laplace transform of each piece is the integral over some subset of $(0, \infty)$, as follows:

$$L[f] = \int_0^{t_1} e^{-st} f_1(t)\ dt + \ldots + \int_{t_{n-1}}^\infty e^{-st} f_n(t)\ dt \tag{1.3}$$

The Second Shift Theorem can be thought of as a special case of this, in which there are two pieces and one of them is the zero function. One example of this in action is the Laplace transform of the unit step function, defined as $g(t) = 1$ for $0 \leq t \leq 1$ and $g(t) = 0$ everywhere else:

$$L[g] = \int_0^1 e^{-st}\ dt = \frac{1}{s}\left(1 - e^{-s}\right) \tag{1.4}$$

Anyway, since this is a course on ODEs, the main reason for learning about Laplace transforms in this course is to use them to solve ODEs. We have previously seen that when you take the Laplace transform of the derivative of a function, the formula reduces to something involving the Laplace transform of the original function:

$$L[f'] = sL[f] - f(0) \tag{1.5}$$

We can use this as a recurrence relation to get the Laplace transform of any derivative in terms of the Laplace transform of the original function. However, since we have only solved ODEs up to second order in this course, we are mostly concerned with the second derivative:

$$L[f''] = sL[f'] - f(0) = s^2L[f] - sf(0) - f'(0) \tag{1.6}$$

We now have all of the tools we need to solve ODEs with this. Let's start with one that we already know the solution for, just to make sure that it works. Suppose we have the initial value problem $y'' + y = 0$, $y(0) = 1$, $y'(0) = 2$. We know that this will have a solution of $y(t) = C_1 \cos t + C_2 \sin t$, and by applying the initial conditions, we get that $C_1 = 1$ and $C_2 = 2$. If we wanted to solve this using Laplace transforms, we would just take the Laplace transform of both sides in order to get the y'' term into a form with just y:

$$L[y''] + L[y] = 0 \tag{1.7}$$

Note that we can separate the two terms inside the Laplace operator, because it is a linear operator. Note also that the Laplace transform of 0 is 0, as mentioned above. Using our formula for $L[y'']$, and using the notation Y for $L[y]$, we get the following:

$$s^2Y - sy(0) - y'(0) + Y = 0 \tag{1.8}$$

The goal now becomes to find Y, as it might be the Laplace transform of some function that we know. To do this, we can substitute in our initial conditions and group like terms:

$$Y\left(s^2 + 1\right) - s - 2 = 0 \implies Y = \frac{s+2}{s^2+1} \tag{1.9}$$

Remember that $L[\cos \omega t] = \frac{s}{s^2+\omega^2}$ and $L[\sin \omega t] = \frac{\omega}{s^2+\omega^2}$. Using 1 for ω and breaking apart our fraction, we get the following:

$$Y = \frac{s}{s^2+1} + 2\frac{1}{s^2+1} = L[\cos t + 2\sin t] \tag{1.10}$$

When we "invert" the Laplace transform that we ended up with, we get exactly the solution that we were looking for.

We can also use this method to solve ODEs with forcing functions. For instance, let's use the example that we just worked through, but with a forcing function of $7e^{2t}$ instead of 0. For this DE, the complementary solution is still $y_c = C_1 \cos t + C_2 \sin t$. The particular solution can be obtained using methods that we already know, and it works out to be $y_p = \frac{7}{5}e^{2t}$. Because we have this extra term, C_1 and C_2 turn out to be slightly different, namely $\frac{-2}{5}$ and $\frac{-4}{5}$. Let's see what happens when we use the Laplace transform on this DE:

$$L[y''] + L[y] = Y\left(s^2+1\right) - s - 2 = L[7e^{2t}] = \frac{7}{s-2} \tag{1.11}$$

When we isolate Y, we get two terms on the right-hand side. One of these is the same as what we had earlier (for the version of this ODE without a forcing function), but the other one is new:

$$Y = \frac{s+2}{s^2+1} + \frac{7}{(s-2)(s^2+1)} \tag{1.12}$$

Since the Laplace transform is linear, we might be tempted to consider each term separately and claim that part of the solution is $\cos t + 2\sin t$, as before. However, the other term isn't the Laplace transform of any function that we know of, so we can't actually do this. Instead, we'll need to break it apart by using partial fractions. In the interest of speed, this part of the derivation will be skipped. The result of the partial fractions is the following:

$$Y = \frac{7}{5}\frac{1}{s-2} - \frac{2}{5}\frac{s}{s^2+1} - \frac{4}{5}\frac{1}{s^2+1} \tag{1.13}$$

We know what each of these terms is the Laplace transform of, so we can easily verify that the solution we got using this method is the same as the expected one.

Using the same techniques, we can apply the method of Laplace transforms to general second-order linear ODEs with constant coefficients. If we have the ODE $ay''+by'+cy = p(t)$ with initial conditions $y(0)$ and $y'(0)$ defined, then we get the following, for Y and P the Laplace transforms of y and p:

$$Y = \frac{as + ay'(0) + by(0)}{as^2+bs+c} + \frac{P(s)}{as^2+bs+c} \tag{1.14}$$

Getting these into forms that we can use will typically involve partial fractions, and may also involve the Shift Theorems. The first term on the right-hand side usually isn't that hard to deal with. There are three separate cases for what you'll get when you attempt to use partial fractions to get it into a usable form. The first case is if as^2+bs+c has two real roots, so $as^2+bs+c = a(s-s_1)(s-s_2)$:

$$\frac{as + ay'(0) + by(0)}{as^2+bs+c} = \frac{k_1}{s-s_1} + \frac{k_2}{s-s_2} \tag{1.15}$$

The second case is if as^2+bs+c has a repeated real root, so $as^2+bs+c = a(s-s_1)^2$:

$$\frac{as + ay'(0) + by(0)}{as^2+bs+c} = \frac{k_1}{s-s_1} + \frac{k_2 s + k_3}{(s-s_1)^2} \tag{1.16}$$

The third case is if as^2+bs+c has complex conjugate roots:

$$\frac{as + ay'(0) + by(0)}{as^2+bs+c} = \frac{k_1 s + k_2}{(s-\frac{b}{2a})^2 + (\frac{c}{a} - \frac{b^2}{4a^2})} \tag{1.17}$$

The term with $P(s)$ is usually the more challenging one. A good first effort would be to use partial fractions and (if need be) the Shift Theorems on it. If that doesn't work, you

could also use the Convolution Theorem. Suppose you have some function $H(s)$ that can be expressed as $F(s)G(s)$ for some F and G. You might think that if $H(s) = L[h(t)]$ and so on, then $h(t)$ would be equal to $f(t)g(t)$. However, this is not necessarily true. Instead of being the product of f and g, we instead get that h is the convolution of f and g. If you haven't seen convolution before, then the convolution of two functions f and g, denoted $f * g$, is a certain kind of integral:

$$f * g = \int_0^t f(u)g(t-u)\ du = \int_0^t f(t-u)g(u)\ du = g * f \tag{1.18}$$

Note that taking the convolution is commutative. It can also be thought of as a "reverse inner product" over function space. With the standard inner product of two functions f and g, we take the integral over all points of the product of f and g evaluated at the same point. With the convolution $f * g$, we're also multiplying a value taken by f with a value taken by g, but the points of f start from the left-hand side of the domain and move right, while the points of g start from the right-hand side of the domain and move left. (This is easiest to visualize when the convolution integral has finite bounds, due to one or both of the functions having a compact support.)

1.2 November 3: More on using Laplace transforms to solve ODEs, including ODEs with piecewise forcing terms

Previously, we saw a way to use Laplace transforms to solve general second-order ODEs with constant coefficients. Depending on the problem, you may find that this method requires more or less work than others (such as variation of parameters). However, there is one class of ODEs that Laplace transforms are particularly useful for solving, namely those with discontinuous forcing functions. This is because the Laplace transform of the entire forcing function can be broken up into multiple pieces (as we saw previously), and we can deal with the pieces using the Shift Theorems.

Let's see a very simple example of this in action. Suppose that we have a first-order ODE with a piecewise continuous forcing function. For instance, we may be interested in calculating the velocity of some object if we apply some constant level of force to it up until a specified time T, and then a different level of force to it for $t > T$. We can get the general form of this ODE from Newton's law:

$$F = ma = m\frac{dv}{dt} \implies \frac{dv}{dt} = \begin{cases} \frac{1}{m}F_1, & 0 \le t \le T \\ \frac{1}{m}F_2, & t > T \end{cases} \tag{1.19}$$

Because the integral of a constant is just a linear function, we can determine the solution of this ODE without any advanced techniques. v will increase linearly at a rate $\frac{F_1}{m}$ from time 0 to time T, and then at a rate $\frac{F_2}{m}$ thereafter. This can be written in the following way, assuming we know the initial condition $v(0)$:

$$v(t) = \frac{F_1}{m}t + \mathrm{H}(t-T)\frac{F_2 - F_1}{m}(t-T) + v(0) \tag{1.20}$$

Here, H represents the Heaviside step function. This function, named after the English applied mathematician Oliver Heaviside, is equal to zero for $t < 0$ and one for $t > 0$; in this particular example, it has been shifted to the right by T units. This has the effect of "turning off" the second term in the equation when $t < T$, making $v(t)$ the much simpler function of $v(t) = \frac{F_1}{m}t + v(0)$. When $t > T$, this formulation for v means that we subtract the $\frac{F_1}{m}t$ term and are left with a function that grows linearly at a rate $\frac{F_2}{m}$. In the second term, we use $t - T$ for time dependence rather than just t, because the line $\frac{F_2 - F_1}{m}t$ passes through the origin instead of the point $(T, v(0) + \frac{F_1}{m}T)$.

Now, let's solve this using Laplace transforms. Before we do this, it's best to get the piecewise forcing function expressed in terms of Heaviside step functions. This can be done like so:

$$\frac{F_1}{m} + \frac{F_2 - F_1}{m}\mathrm{H}(t-T) = \begin{cases} \frac{1}{m}F_1, & 0 \le t \le T \\ \frac{1}{m}F_2, & t > T \end{cases} \tag{1.21}$$

Just like before (when we solved this ODE by inspection), we use a Heaviside step function to "turn on" F_2 and "turn off" F_1 at $t = T$. Now, we can take the Laplace transform of both sides, keeping in mind that it is a linear operator and hence we can pull out constants:

$$L[v'] = sL[v] - v(0) = \frac{F_1}{m}L[1] + \frac{F_2 - F_1}{m}L[\mathrm{H}(t-T)\cdot 1] \tag{1.22}$$

The first term on the right-hand side is easy to evaluate; remember that $L[1] = L[t^0]$, which we have a formula for. In the second term on the right-hand side, we are taking the Laplace transform of a shifted version of the constant function $f(t) = 1$. Thus, we can use the Second Shift Theorem to evaluate that term. We end up with the following, for $V = L[v]$:

$$sV - v(0) = \frac{F_1}{m}\frac{1}{s} + \frac{F_2 - F_1}{m}\frac{1}{s}e^{-Ts} \tag{1.23}$$

Isolating V gives us the following:

$$V = \frac{F_1}{ms^2} + \frac{F_2 - F_1}{ms^2}e^{-Ts} + \frac{v(0)}{s} \tag{1.24}$$

We know that $\frac{1}{s^2}$ is the Laplace transform of $t^1 = t$. Additionally, because of the e^{-Ts} in the middle term, we know that everything in that term will be shifted by T. (We will once again use the Heaviside step function to represent the outcome of this shift, so that the term is only nonzero for values of t larger than whatever the origin was shifted to, in this case $t = T$.) We can therefore find v, which takes exactly the same form as our expected answer:

$$v = \frac{F_1}{m}t + \frac{F_2 - F_1}{m}(t-T)\mathrm{H}(t-T) + v(0) \tag{1.25}$$

If we wanted to, we could also integrate another time to find the position function. The math for this is mostly the same, although we get another factor of s that we need to divide through by (from the relation that $L[x'] = sL[x]$). This gives us the following:

$$X = \frac{F_1}{ms^3} + \frac{F_2 - F_1}{ms^3}e^{-Ts} + \frac{v(0)}{s^2} + \frac{x(0)}{s} \tag{1.26}$$

When solving this, remember that $L[t^n] = \frac{n!}{s^{n+1}}$. This means that $\frac{1}{s^3}$ is the Laplace transform of $\frac{1}{2}x^2$, not x^2, so we get an extra factor of $\frac{1}{2}$ when inverting the Laplace transform. This yields the following result, as expected:

$$x = \frac{F_1}{2m}t^2 + \frac{F_2 - F_1}{2m}(t-T)^2\mathrm{H}(t-T) + v(0)t + x(0) \tag{1.27}$$

Before we dive into another example of solving ODEs using Laplace transforms, I'll take a quick intermission to explain three very important functions in applied math, which are linked by being the derivatives of each other. The first of these is the ramp function. This is defined as follows:

$$\mathrm{R}(t) = \max(0, t) = \begin{cases} 0, & t < 0 \\ t, & t \geq 0 \end{cases} \tag{1.28}$$

For positive input, the ramp function increases linearly, while for negative input it is flat. This makes its graph resemble a ramp coming out of the ground, leading to its name. Many of the applications of the ramp function come in machine learning. There, it is referred to as the ReLU function (for "rectified linear unit"), and within a neural network it serves to determine which inputs are propagated forward through the network and which are ignored. In this way, it is responsible for the "learning" aspect of machine learning.

The Heaviside step function is the derivative of the ramp function. It was originally developed in electrical engineering to represent the process of switching on a circuit, and can be used more generally in the modelling of any signal that is active at some times and inactive at others. (By scaling and combining Heaviside functions, we can specify the strength of the signal as well as exactly which times it is on and off.) It is formally defined as follows:

$$\mathrm{H}(t) = \begin{cases} 0, & t < 0 \\ 1, & t > 0 \end{cases} \tag{1.29}$$

Note that the value that the Heaviside step function takes at 0 is not defined. This is because $\mathrm{H}(0)$ can take a few different values, depending on whether any specific property of H is desired. For instance, if we want H to be equal to its limit at 0 from above or from below, we would take $\mathrm{H}(0)$ to be 1 or 0, respectively. $\frac{1}{2}$ is also a common choice, as it makes H an odd function, as well as serving as an analogue to the half-saturation constant in a logistic function (which H resembles).

Regardless of which value is chosen, the Heaviside step function is vertical at $t = 0$, so it has an infinite slope there. However, it is flat everywhere else. This means that taking its derivative would result in a function defined like this:

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases} \tag{1.30}$$

Alternatively, we could say that this function is undefined at 0. We additionally have the constraint that the derivative of the Heaviside step function integrates to 1, as $\mathrm{H}(t) = 1\ \forall t > 0$:

$$\int_{-\infty}^{\infty} \delta(t)\ dt = 1 \tag{1.31}$$

The function that satisfies both of these constraints is the Dirac delta function. It is named after Paul Dirac, who introduced it in his 1930 textbook "The Principles of Quantum Mechanics", one of the first on that subject. On the surface, the Dirac delta function appears rather abstract. So, I'll show you an example of how it can be derived, as well as a problem in ODEs where it can be used. Suppose that we have some amount of a radioactive chemical in a beaker, which will decay at a rate k. This can be modelled as $\frac{dx}{dt} = -kx$ for x the amount of chemical present, as we have seen before. Now, suppose that for some length of time starting at $t = a$ and continuing for Δ units of time, we add more of the chemical into the beaker. Suppose also that we add the chemical at a constant rate, and the total amount of chemical that we add is b. This means that our new ODE describing the amount of chemical in the beaker is as follows:

$$\frac{dx}{dt} = \begin{cases} -kx,\ t < a \\ -kx + \frac{b}{\Delta},\ a \leq t \leq a + \Delta \\ -kx,\ t > a + \Delta \end{cases} \tag{1.32}$$

At any given point in time during the interval $[a, \Delta]$, the rate that we add the chemical is $\frac{b}{\Delta}$. This is because the total amount added is b, so if we integrate just the amount added from a to Δ (ignoring the decay for a moment), we must get b. This can be summarized by the following equation:

$$\int_{a}^{a+\Delta} \frac{b}{\Delta}\ dt = \frac{b}{\Delta}\left(a + \Delta - a\right) = b \tag{1.33}$$

Within the context of the original ODE, we can use Heaviside step functions to express the process of adding the chemical as one that gets turned on and then back off. Specifically, we can use a positive Heaviside step function to represent the start of adding the chemical, and a negative one to represent the end of it. Using f to denote the rate at which the chemical is added, we get the following:

$$f(t) = \frac{b}{\Delta}\left(\mathrm{H}(t - a) - \mathrm{H}(t - (a + \Delta))\right) \tag{1.34}$$

We now have $\frac{dx}{dt} = -kx + f(t)$, and we can solve this using Laplace transforms. If we take the Laplace transform of both sides and isolate $X = L[x]$, then we get the following (for $F = L[f]$):

$$X = \frac{x(0)}{s + k} + \frac{F}{s + k} \tag{1.35}$$

The first term is relatively easy to invert, since $\frac{1}{s+k}$ is the Laplace transform of e^{-kt}. The second one needs to be handled using the Convolution Theorem, since it is of the form

$H(s) = F(s)\frac{1}{s+k}$. We therefore get the following convolution integral that we need to evaluate in order to invert the second term:

$$f * e^{-kt} = \int_0^t f(u)e^{-k(t-u)}\ du \tag{1.36}$$

This needs to be broken up into two different parts, on account of the two Heaviside step functions that make up f. After some integration, we get the following for our solution x:

$$x = x(0)e^{-kt} + \frac{b}{k\Delta}\left(1 - e^{-k(t-a)}\right)\mathrm{H}(t-a) + \frac{b}{k\Delta}\left(1 - e^{-k(t-(a+\Delta))}\right)\mathrm{H}(t-(a+\Delta)) \tag{1.37}$$

However, what happens when we add the chemical over a smaller and smaller interval? In other words, what happens when we decrease Δ so that it goes to 0? As Δ shrinks, the rate of input ($\frac{b}{\Delta}$) gets arbitrarily large. However, we are still adding b units of the chemical no matter how small the interval gets, as the integral describing how much chemical is added does not change at all and still equals b. In other words, the limit as Δ approaches zero of the rate of input is undefined, but the limit as Δ approaches zero of how much chemical is added is b. This results in f becoming a rather paradoxical mathematical object that is undefined at a single point (escaping to $+\infty$ there) and zero everywhere else, but whose definite integral across its entire domain is equal to the finite value b. This is the Dirac delta function. If a is taken to be 0 and b is taken to be 1, then it becomes clear that the integral of this function is the Heaviside step function $\mathrm{H}(t)$. Choosing different values for a and b causes the integral to be scaled and shifted, with the result of $b\mathrm{H}(t-a)$. A method of defining the Dirac delta function that reflects its status as a limit is shown below:

$$I_\varepsilon(t) = \begin{cases} \frac{1}{\varepsilon},\ 0 \le t \le \varepsilon \\ 0,\ t > \varepsilon \end{cases} \implies \delta(t) = \lim_{\varepsilon \to 0} I_\varepsilon(t) \tag{1.38}$$

So, what is the Laplace transform of the Dirac delta function? Well, for any continuous function $g(t)$ as well as $I_\varepsilon(t)$ as defined above, we have the following result:

$$\int_{-\infty}^{\infty} I_\varepsilon(t)g(t)\ dt = \int_0^\varepsilon I_\varepsilon(t)g(t)\ dt = \frac{1}{\varepsilon}\int_0^\varepsilon g(t)\ dt \tag{1.39}$$

By the Mean Value Theorem for Integrals, we also have that within the interval $[0, \varepsilon]$ there exists some c such that the following holds:

$$\int_0^\varepsilon g(t)\ dt = \varepsilon \cdot g(c) \implies \int_0^\varepsilon I_\varepsilon(t)g(t)\ dt = g(c) \tag{1.40}$$

If we take the limit as ε goes to 0 (approaching the Dirac delta function), we get that c also goes to 0, since 0 eventually becomes the only value in the interval $[0, \varepsilon]$. Therefore, we get the following:

$$\lim_{\varepsilon \to 0} \int_0^\varepsilon I_\varepsilon(t)g(t)\ dt = g(0) \tag{1.41}$$

This can, of course, be shifted to $g(a)$ for any a, if the Dirac delta function is centered there instead of at 0. Since exponential functions are continuous, we can combine a few of the above results to get a statement for the Laplace transform of the Dirac delta function:

$$L[\delta(t)] = \int_0^\infty e^{-st}\delta(t)\ dt = e^{-s\cdot 0} = 1 \tag{1.42}$$

For a Dirac delta function centered at some $a \neq 0$, we get the following result instead:

$$L[\delta(t)] = \int_0^\infty e^{-st}\delta(t-a)\ dt = e^{-as} \tag{1.43}$$

This is consistent with the Second Shift Theorem, since the result can also be thought of as e^{-as} times the Laplace transform of the "unshifted" Dirac delta function (which is 1). Now that we know this, we can solve the ODE modelling the chemical in a beaker that we talked about earlier with an instantaneous addition of the chemical. The ODE now takes the following form:

$$\frac{dx}{dt} = -kx + b\delta(t-a) \tag{1.44}$$

Taking the Laplace transform of both sides gives us this:

$$sX - x(0) = -kX + be^{-as} \tag{1.45}$$

Isolating X produces the following:

$$X = \frac{x(0)}{s+k} + \frac{b}{s+k}e^{-as} \tag{1.46}$$

The first term is the Laplace transform of an exponential, while the second one is an exponential that has been shifted. We can invert and get the following solution for x:

$$x(t) = x(0)e^{-kt} + be^{-k(t-a)}\mathrm{H}(t-a) \tag{1.47}$$

We end up with two exponential decay curves that produce a jump discontinuity when graphed, as we might expect given how the problem was formulated.

Chapter 2

Systems of differential equations

2.1 November 5: Introduction to systems of differential equations, including a review of linear algebra

Previously, we've focused on solving single differential equations. However, in practice, many things that you may want to model using differential equations will interact with each other. This means that we will need to consider solving multiple differential equations at the same time, with each of the differential equations in question representing the rate of change of some quantity that we are interested in. For instance, suppose that we are modelling a chemical reaction where one reactant is converted into one product, which then decays:

$$A \longrightarrow B \longrightarrow \tag{2.1}$$

From this reaction schematic, we can determine a simple differential equation model of the concentrations of both A and B. We assume that A is being converted into B at a linear rate (as this is a first-order reaction), and that it is not being produced by any mechanism. This leads to the following dynamics, which we have seen a few times in this course:

$$\frac{dA}{dt} = -k_1 A(t) \tag{2.2}$$

On the other hand, there are two rates associated with B, namely the rate at which it is produced and the rate at which it decays. The decay rate of B is easily modelled, using the same techniques as we've been through before. However, the production rate of B depends on how much A is available, rather than being a function of B. More specifically, it is the opposite of the corresponding term in $\frac{dA}{dt}$, due to the reaction stoichiometry (where one molecule of B is produced for every one molecule of A consumed). This leads to the following differential equation for B:

$$\frac{dB}{dt} = k_1 A(t) - k_2 B(t) \tag{2.3}$$

Note that integrating $\frac{dB}{dt}$ requires knowledge of $A(t)$, so we can't solve for B on its own. Instead, we need to consider both $\frac{dA}{dt}$ and $\frac{dB}{dt}$ together, in what is called a system of differential equations. You may see these written like this:

$$\begin{cases} \frac{dA}{dt} = -k_1 A \\ \frac{dB}{dt} = k_1 A - k_2 B \end{cases} \tag{2.4}$$

Note that this is written in a slightly different form than what you have been seeing in other parts of this course, with the time derivative isolated on the left-hand side rather than the forcing function (if any) isolated on the right-hand side. This is because systems of differential equations often come up in mathematical models, where we know the rate of change of some quantity and are interested in determining how that quantity evolves over time.

Systems of differential equations can arise in other ways besides the interactions of several different variables. Another way that you can get a system is by taking a single differential equation of higher than first order and defining extra variables to make it easier to solve (or simulate). You can do this with differential equations of any order, although in practice, third-order ODEs will usually be the highest that you'll see. One example where you might see a third derivative is the process of controlling an elevator, or any other object that starts at rest, accelerates to some velocity, then returns to rest upon reaching its destination. As the acceleration changes over time, its derivative must be nonzero. In fact, limiting this third derivative of position (referred to as jerk) is very important, since high values of jerk may cause passengers to feel uncomfortable.

For a straightforward example of turning a higher-order ODE into a system of first-order ones, suppose we have a third-order ODE with constant coefficients:

$$x''' + ax'' + bx' + cx = 0 \tag{2.5}$$

We haven't learned how to solve these using our previous methods. However, we can define the quantities $y = x'$ and $z = y' = x''$ in order to get three first-order differential equations, like this:

$$\begin{cases} x' = y \\ y' = z \\ z' = -az - by - cx \end{cases} \tag{2.6}$$

This is actually quite straightforward to solve, which you can do using linear algebra. Before we learn how to do this, I'll do a quick review of some linear algebra basics. First, there's the concept of linear independence, which you have seen earlier in this course with regards to finding solutions to second-order ODEs. In the context of linear algebra, linear independence is similar. If we have vectors $v_1, \ldots, v_n$, then these vectors are linearly dependent if at least one of them can be written as a scalar multiple of the others, and linearly independent if this cannot be done. In other words, $v_1, \ldots, v_n$ are linearly independent if the following holds:

$$a_1 v_1 + \ldots + a_n v_n = 0 \implies a_i = 0 \ \forall i \tag{2.7}$$

For some easy examples, $v_1 = [1\ 0]^T$ and $v_2 = [1\ 1]^T$ are linearly independent, whereas $w_1 = [1\ 0\ 1]^T$, $w_2 = [1\ 1\ 0]^T$ and $w_3 = [0\ -1\ 1]^T$ are linearly dependent because w_3 can be expressed as $w_1 - w_2$.

If you have a set of vectors that make up the rows or columns of a matrix, then one way to determine if they are linearly independent is to take the determinant of the matrix. The determinant of a matrix is defined recursively. For a 1x1 matrix, it's just the single value in the matrix. For a matrix with higher dimensions, calculating the determinant is done by going along one particular row (or down one particular column) and taking the sum of the entries in that row or column multiplied by what are called the cofactors in that row or column. For each entry a_{ij} in a matrix A, the cofactor C_{ij} associated with that entry is defined in the following way, where M_{ij} is the matrix formed by deleting row i and column j from A (also known as a minor):

$$C_{ij} = (-1)^{i+j} \det M_{ij} \tag{2.8}$$

So, the determinant is just $a_{1,1}C_{1,1} + a_{1,2}C_{1,2} + \ldots + a_{1,n}C_{1,n}$ for an $n \times n$ matrix, or equivalently $a_{2,1}C_{2,1}+a_{2,2}C_{2,2}+\ldots+a_{2,n}C_{2,n}$, or $a_{1,1}C_{1,1}+a_{2,1}C_{2,1}+\ldots+a_{n,1}C_{n,1}$ if expanding along the first column, et cetera. One trick to doing this is to expand along a row or column that has 0 for some of its entries, as that makes the computation easier. If this determinant is nonzero, then the row vectors or column vectors making up the matrix are linearly independent.

Another important concept is invertibility of a matrix. If a matrix A is invertible, then there exists some other matrix A^{-1} for which A times A^{-1} is equal to the identity matrix I. (Remember that the identity matrix I has the property that $AI = A$ for any other matrix A, and is defined by having ones on its diagonal and zeros everywhere else.) If the determinant of a matrix is nonzero, then it is invertible. If the determinant is zero, then its inverse does not exist, and it is referred to as singular.

In general, multiplying a matrix by a vector results in another vector. In some cases, this operation may be equivalent to multiplying the original vector by a constant (in other words, scaling each element of the vector by the same amount). For a given matrix A, the vectors x that can be operated on in this manner satisfy the following property, for some constant λ:

$$Ax = \lambda x \tag{2.9}$$

In this case, x is called an eigenvector of A, and λ is its associated eigenvalue. We can find these by manipulating the above equation to get the following:

$$(A - \lambda I)x = 0 \tag{2.10}$$

Note that $(A - \lambda I)$ is a matrix, and that it needs to have a determinant of zero in order for $(A - \lambda I)x = 0$ to have nonzero solutions for x. This is because the kernel of $(A - \lambda I)$ (i.e. the set of vectors that $(A - \lambda I)$ maps to the zero vector) needs to be non-trivial, which cannot happen if all of the rows of $(A - \lambda I)$ are linearly independent as a consequence of the rank-nullity theorem. Therefore, we need to find λ that satisfies the following equation:

$$\det (A - \lambda I) = 0 \tag{2.11}$$

In order to do this, remember that λI is just the identity matrix multiplied by the scalar λ, so $(A - \lambda I)$ will necessarily have diagonal entries of the form $a_{ii} - \lambda$. Since λ is

unknown, this means that the determinant of $(A - \lambda I)$ will be a polynomial in λ rather than a single constant. This polynomial is called the characteristic equation of A, and solving it for λ produces the eigenvalues of A. Once the eigenvalues have been found, finding the corresponding eigenvectors can be done by substituting each eigenvalue back into the matrix equation $(A - \lambda I)x = 0$ and solving the resulting equation for x. (Since finding the determinant is easier if you expand along a row or column that is mostly zeros, you might expect there to be some trick involving doing that when finding eigenvalues, and indeed there is. The eigenvalues of a triangular matrix, one where all of the entries above or below the main diagonal are zero, are equal to the diagonal entries of that matrix.)

Here's an example of how to find eigenvalues and eigenvectors. Suppose that we have the following matrix:

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \tag{2.12}$$

Subtracting λI from this matrix, i.e. subtracting λ from each of the diagonal entries, gives us the following:

$$A = \begin{bmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix} \tag{2.13}$$

We will now expand along the first row to obtain the determinant of this matrix. This produces a quadratic polynomial for λ, as this is a 2×2 matrix:

$$\det A - \lambda I = (1-\lambda)(1-\lambda) - 4 = \lambda^2 - 2\lambda - 3 = (\lambda - 3)(\lambda + 1) \tag{2.14}$$

This factors rather nicely, and gives us the eigenvalues of $\lambda = 3$ and $\lambda = -1$. Now, it remains to find their corresponding eigenvectors. Consider the equation $(A - \lambda I)x = 0$ for the case where $\lambda = 3$:

$$\begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix} x = 0 \tag{2.15}$$

If we assume that x is a vector with two entries, or $x = [x_1 \; x_2]^T$, then both the first and second rows of the matrix in this matrix-vector equation imply that $x_1 = x_2$. We can thus assume that the eigenvector associated with $\lambda = 3$ is any vector satisfying these requirements. (For instance, we can go with $[1 \; 1]^T$.) Considering the case where $\lambda = -1$ gives us a different equation to solve:

$$\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} x = 0 \tag{2.16}$$

Here, we get the condition that $x_1 = -x_2$. Therefore, we can choose $[-1 \; 1]^T$ for the eigenvector associated with $\lambda = -1$, although once again any multiple of this is also an eigenvector. Note that in this case, the eigenvalues were distinct. However, it is possible to have repeated roots of the characteristic polynomial, and therefore repeated eigenvalues. One very simple example of this is the identity matrix I. When we subtract λ from the diagonal and take the determinant, we get the following:

$$\begin{vmatrix} 1-\lambda & 0 \\ 0 & 1-\lambda \end{vmatrix} = (1-\lambda)^2 \implies \lambda = 1 \tag{2.17}$$

We call $\lambda = 1$ an eigenvalue of multiplicity 2. Note however that I still has two distinct eigenvectors, which we can once again get by solving $(A - \lambda I)x = 0$ for $\lambda = 1$. When doing this, we get the equation $0x = 0$ and can therefore choose any vectors for x. The canonical ones are $[1\ 0]^T$ and $[0\ 1]^T$. These together span all of $\mathbb{R}^2$, which is intuitive as I has the property that $Ix = x$ for any vector x (and hence all vectors in $\mathbb{R}^2$ are eigenvectors of I).

2.2 November 8: Solving systems of linear, homogeneous first-order ODEs with constant coefficients

As with uncoupled ODEs, we'll start our journey into systems of ODEs with the simplest possible case, primarily because it's one of the few for which there exist methods of solving analytically. More specifically, we'll start with linear, homogeneous first-order ODEs with constant coefficients. Remember at the very beginning of this course that we determined that the solution to the ODE $x' = kx$ is $x(t) = Ce^{kt}$. Suppose that we take a similar system:

$$\begin{cases} \frac{dx_1}{dt} = ax_1 + bx_2 \\ \frac{dx_2}{dt} = cx_1 + dx_2 \end{cases} \tag{2.18}$$

The solutions for $x_1(t)$ and $x_2(t)$ will also contain exponential functions. However, because each ODE in this system has two terms, each affecting the growth or decay of x_1 or x_2 over time, a single exponential function might not be enough to describe the dynamics of the two state variables. Instead, the solutions for x_1 and x_2 will each be a linear combination of two different exponential functions. An exception to this is if one of the state variables is uncoupled, i.e. its rate of change only depends on itself (and not any of the other state variables). (In the above system, that would mean that $b = 0$ and/or $c = 0$.) In this case, the ODE for that state variable can be solved independently of the rest of the system.

To solve this system of differential equations, we will need to rewrite it as a matrix. We can construct a vector whose entries are the state variables of the system, like this:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.19}$$

Using this vector-valued function, as well as some linear algebra, we can rewrite our system of ODEs as a matrix-vector equation:

$$\mathbf{x}' = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \mathbf{x} = \mathbf{M}\mathbf{x} \tag{2.20}$$

We will now assume that our vector-valued function $\mathbf{x}$ will consist of exponential functions. In other words, for some vector $\mathbf{u}$ that we will eventually solve for the entries of, and some constant r that we will also eventually find, we will assume the following:

$$\mathbf{x} = \mathbf{u}e^{rt} \implies \mathbf{x}' = r\mathbf{u}e^{rt} \tag{2.21}$$

Plugging this into our original system, we get the following:

$$r\mathbf{u}e^{rt} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \mathbf{u}e^{rt} = \mathbf{M}\mathbf{u}e^{rt} \tag{2.22}$$

Because e^{rt} is nonzero, we can divide both sides by it. We then can rearrange terms to get our equation into the following form:

$$(\mathbf{M} - r\mathbf{I})\mathbf{u} = 0 \tag{2.23}$$

This looks very similar to the setup of an eigenvalue problem so far, which isn't a coincidence. Continuing with the theme, we need $\mathbf{M} - r\mathbf{I}$ to have a nontrivial kernel in order to get some solutions other than $\mathbf{u} = 0$. This occurs when the matrix $\mathbf{M} - r\mathbf{I}$ has a determinant of zero, so we will need to solve for the values of r that make it so. In this case, finding the determinant of a 2×2 matrix is straightforward:

$$\det(\mathbf{M} - r\mathbf{I}) = \begin{vmatrix} a - r & b \\ c & d - r \end{vmatrix} = (a - r)(d - r) - bc = r^2 + (-a - d)r + (ad - bc) \tag{2.24}$$

This produces a polynomial in r that can be solved. This is indeed the characteristic polynomial for the matrix $\mathbf{M}$, meaning that we need to find the eigenvalues of $\mathbf{M}$ as part of finding the solution to our system. In this specific case, since we have assumed a 2×2 system, we will have at most 2 distinct eigenvalues, and therefore at most 2 values of r. Continuing this process, we still need to solve for $\mathbf{u}$ for each value of r, which is equivalent to finding the eigenvectors of $\mathbf{M}$. Once we do this, then as we assumed that $\mathbf{x} = \mathbf{u}e^{rt}$ earlier, we will have all of the information necessary to construct a solution $\mathbf{x}$. (Note how I say "a solution" instead of "the solution". This is because there will be multiple eigenvalues and eigenvectors and hence multiple solutions; I'll elaborate on this later.)

In order to illustrate this, I will work through a concrete example. Suppose we have the following system:

$$\begin{cases} \frac{dx_1}{dt} = x_1 + 2x_2 \\ \frac{dx_2}{dt} = 3x_1 + 2x_2 \end{cases} \tag{2.25}$$

We can turn this system into a matrix $\mathbf{M}$, and find its eigenvalues and eigenvectors:

$$\mathbf{M} = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} \implies \det(\mathbf{M} - r\mathbf{I}) = \begin{vmatrix} 1 - r & 2 \\ 3 & 2 - r \end{vmatrix} = r^2 - 3r - 4 \tag{2.26}$$

The characteristic polynomial factors into $(r - 4)(r + 1)$, which means that we have eigenvalues of $r = 4$ and $r = -1$. Now, we will plug in these values to $\mathbf{M} - r\mathbf{I}$ in order to find the eigenvectors of $\mathbf{M}$. Let's start with $r = 4$:

$$r = 4 \implies \begin{bmatrix} -3 & 2 \\ 3 & -2 \end{bmatrix} \mathbf{u} = \begin{bmatrix} -3 & 2 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0 \tag{2.27}$$

When solving this system for u_1 and u_2, we get the relation that $3u_1 = 2u_2$. This points to $[2\ 3]^T$ being an eigenvector with the associated eigenvalue of $r = 4$. Now, we will plug in $r = -1$:

$$r = -1 \implies \begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0 \tag{2.28}$$

This brings us to $u_1 = -u_2$, and hence an eigenvector of $[-1\ 1]^T$ (or any scalar multiple thereof, of course). We now have two solutions to the system, each one of the following form:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{u}e^{rt} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} e^{rt} \tag{2.29}$$

In the specific case that we are dealing with, we have the following two solutions:

$$\mathbf{x}^{(1)} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} e^{4t};\ \mathbf{x}^{(2)} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} e^{-t} \tag{2.30}$$

What can we do with these? Well, as with finding solutions of second-order differential equations, our general solution will just be a linear combination of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. (This is referred to as the principle of superposition. It can be easily proved by taking the derivatives of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, plugging them into the original system, and determining what cancels.) As a result of all this, we can get the following general solutions for x_1 and x_2 in the case that we're looking at. For C_1 and C_2 arbitrary constants, which can be obtained by applying initial conditions, we have:

$$\begin{cases} x_1 = 2C_1e^{4t} - C_2e^{-t} \\ x_2 = 3C_1e^{4t} + C_2e^{-t} \end{cases} \tag{2.31}$$

Note that we have two basis functions that make up our solution for each state variable, just like we did when we were solving second-order ODEs. This is not an accident. Remember that for an ODE of the form $y'' + ay' + by = 0$, we can make the substitution $z = y'$ and get the following system:

$$\begin{cases} y' = z \\ z' = -by - az \end{cases} \tag{2.32}$$

Here, the solutions for both y and z will consist of two basis functions each, which can be found using the same method as we went through above.

When we go through this process for finding solutions, it is important that the solutions that we come up with are linearly independent. Any set of solutions of a system of differential equations in which all of the individual solutions are linearly independent over some interval is called a fundamental set of solutions for that system over that interval. Determining that some solutions of a system of ODEs form a fundamental set of solutions can be done by evaluating the Wronskian of the functions. However, if the ODEs in the system are linear and homogeneous with constant coefficients, then the solutions will all be exponentials and hence evaluating the Wronskian becomes very easy.

In fact, for a given set of solutions to a linear, homogeneous system of ODEs over some interval, the Wronskian will either be identically zero over the entire interval or will never be zero at any point in the interval. This holds regardless of what the coefficients are; they

need not be constants. This result was first proved by Niels Henrik Abel, and therefore is one of the many results that are referred to as "Abel's theorem".

In the example that I showed you in this lecture, everything worked out well, since the matrix $\mathbf{M}$ representing the specific system we were attempting to solve had eigenvalues that were real and distinct. However, not all matrices (or even all real-valued matrices) have this property. Of the ones that don't, there are two main categories that they can fall into. One possibility for real-valued matrices is that we have a repeated eigenvalue, such as in the following system:

$$\begin{cases} \frac{dx_1}{dt} = x_1 - x_2 \\ \frac{dx_2}{dt} = x_1 + 3x_2 \end{cases} \tag{2.33}$$

The other is that the eigenvalues are complex. In real-valued matrices, this means that they will be complex conjugates, but if we have a complex-valued matrix, the eigenvalues can be any complex number. In this system, for instance, both eigenvalues are purely imaginary:

$$\begin{cases} \frac{dx_1}{dt} = -x_1 + 3x_2 \\ \frac{dx_2}{dt} = -2x_1 + x_2 \end{cases} \tag{2.34}$$

As it turns out, solving each of these cases requires some of the same techniques that you have previously applied when finding solutions to second-order ODEs with constant coefficients, because of our assumption (as always) that the solutions are exponentials of some kind. This means that these cases are actually easier to solve than you might think!

2.3 November 10: Solving systems of ODEs with repeated or complex eigenvalues, or with forcing terms

In the previous lecture, we looked at an easy example of a system of linear, homogeneous first-order ODEs with constant coefficients. However, in that case, everything worked out about as well as it possibly could have due to the eigenvalues being real and distinct. What if instead we have complex eigenvalues? For instance, consider the following system:

$$\begin{cases} \frac{dx_1}{dt} = -x_1 + 3x_2 \\ \frac{dx_2}{dt} = -2x_1 + x_2 \end{cases} \tag{2.35}$$

In order to solve this, we first need to find the eigenvalues and eigenvectors of the matrix corresponding to this system. We know how to do this already:

$$\mathbf{M} = \begin{bmatrix} -1 & 3 \\ -2 & 1 \end{bmatrix} \implies \det(\mathbf{M} - r\mathbf{I}) = (-1 - r)(1 - r) + 6 \tag{2.36}$$

A closer look at the characteristic polynomial of our matrix $\mathbf{M}$, or indeed the entries of $\mathbf{M}$, should reveal something important. Because the diagonal entries of $\mathbf{M}$ are 1 and -1, we will get terms of $+r$ and $-r$, which cancel each other out. Additionally, the constant term in the characteristic polynomial will be positive. (From the previous lecture, we know the formula for all of the coefficients of the characteristic polynomial of a 2×2 matrix. The

constant term is $m_{1,1}m_{2,2} - m_{1,2}m_{2,1}$, which in this case is $-1 + 6 = 5$.) This means that we can get our characteristic polynomial into the form $r^2 = -5$, which yields the purely imaginary roots of $r = \pm i\sqrt{5}$. As we're still assuming that our solutions will be exponentials of some kind, this means that we will get some linear combination of $e^{i\sqrt{5}t}$ and $e^{-i\sqrt{5}t}$ for solutions. These are, of course, both wavefunctions.

Since we have two eigenvalues, r_1 and r_2, that are complex conjugates, the corresponding eigenvectors $\mathbf{u}^1$ and $\mathbf{u}^2$ will also be complex conjugates. We can see this by taking the complex conjugate of the equation $(\mathbf{M} - r_1\mathbf{I})\mathbf{u}^{(1)} = 0$, which we know is satisfied because r_1 is an eigenvalue with associated eigenvector $\mathbf{u}^1$. This results in the following relation:

$$\overline{(\mathbf{M} - r_1\mathbf{I})\mathbf{u}^{(1)}} = 0 \implies (\mathbf{M} - \overline{r_1}\mathbf{I})\overline{\mathbf{u}^{(1)}} = 0 \tag{2.37}$$

This is true because $\mathbf{M}$ and $\mathbf{I}$ are real-valued (by our assumption), so they are their own complex conjugates. Hence, because $\overline{r_1} = r_2$, the eigenvector associated with r_2 is the complex conjugate of that associated with r_1. This makes the process of finding two linearly independent solutions much easier.

Let's find the eigenvector associated with $r_1 = i\sqrt{5}$. Plugging this into our system yields the following:

$$\begin{bmatrix} -1 - i\sqrt{5} & 3 \\ -2 & 1 - i\sqrt{5} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0 \tag{2.38}$$

We therefore get the following relations for u_1 and u_2:

$$\begin{cases} (1 + i\sqrt{5})u_1 = 3u_2 \\ 2u_1 = (1 - i\sqrt{5})u_2 \end{cases} \tag{2.39}$$

From the second of these equations, we can obtain the eigenvector $\mathbf{u}^{(1)} = \left[\frac{1}{2}(1 - i\sqrt{5})\ 1\right]^T$, which has the associated eigenvalue $r = i\sqrt{5}$. If instead we take $r = -i\sqrt{5}$, then we will get a vector for $\mathbf{u}^{(2)}$ in which every entry is the complex conjugate of the corresponding entry in $\mathbf{u}^{(1)}$. Specifically, this is $\left[\frac{1}{2}(1 + i\sqrt{5})\ 1\right]^T$. Now that we know r_1, r_2, $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$, we can get a general solution for this problem in the same way as if the eigenvalues were both real.

As a matter of fact, if the eigenvalues are complex conjugates, we can even get real-valued solutions. Suppose that we have some system of ODEs where the characteristic polynomial has complex roots, so that the equation $(\mathbf{M} - r\mathbf{I})\mathbf{u} = 0$ will feature complex values for r and the entries of $\mathbf{u}$. In other words, if we have $\mathbf{x} = \mathbf{u}e^{rt}$, we can assume that $\mathbf{x} = (\mathbf{v} + i\mathbf{w})e^{(\alpha+i\beta)t}$, where $\mathbf{v}$ and $\mathbf{w}$ have real elements, and α and β are both real constants. After expanding everything out, we therefore get the following solution:

$$\mathbf{x} = e^{\alpha t}(\mathbf{v}\cos\beta t - \mathbf{w}\sin\beta t) + ie^{\alpha t}(\mathbf{v}\sin\beta t + \mathbf{w}\cos\beta t) \tag{2.40}$$

However, this is just one solution. There will be two, and the second one will have values for $\mathbf{u}$ and r that are the complex conjugates of what we found above. As a matter of fact, by taking scalar multiples of both of these solutions and adding them together, we get that both of the following are solutions for $\mathbf{x}$:

$$\mathbf{x} = e^{\alpha t}(\mathbf{v}\cos\beta t - \mathbf{w}\sin\beta t) \tag{2.41}$$

$$\mathbf{x} = e^{\alpha t}(\mathbf{v}\sin\beta t + \mathbf{w}\cos\beta t) \tag{2.42}$$

This, once again, is similar to what we found for second-order linear ODEs with constant coefficients. The functions above are linearly independent (this can be checked using the Wronskian), and we can use them as a basis for the general solution. (This basis is actually preferred, since the functions involved are real-valued.)

So, we have solved the case of a 2×2 system of ODEs with real, constant coefficients where the eigenvalues are complex. What about higher-dimensional systems? Well, in a 3×3 system of ODEs (once again, with real, constant coefficients), you could potentially have a matrix where one of the eigenvalues is real and the other two are complex conjugates of each other. In this case, you can find the solution associated with the real eigenvalue as you normally would, and the two solutions associated with the complex eigenvalues using the method above. For 4×4 systems, and those whose dimensionality is even higher, you could have multiple different complex conjugate pairs, and hence you would have to do the above method multiple times.

What about if the matrix that we form out of our system has repeated eigenvalues? In some cases, that may not matter. For instance, in any $n \times n$ matrix that has repeated eigenvalues but is real-valued and symmetric, we will have n linearly independent eigenvectors regardless. (A symmetric matrix is one that is equal to its own transpose. In other words, if $\mathbf{M}$ is the matrix, then $m_{ij} = m_{ji}\ \forall i, j$.) One example of this is the identity matrix $\mathbf{I}$, which we saw previously has an eigenvalue of multiplicity 2 but two linearly independent eigenvectors.

What if the matrix is not symmetric? Then, we need to find another linearly independent solution somehow. I will illustrate this with an example. Suppose we have the following system:

$$\begin{cases} \frac{dx_1}{dt} = x_1 - x_2 \\ \frac{dx_2}{dt} = x_1 + 3x_2 \end{cases} \tag{2.43}$$

Using what we already know, we can get $r = 2$ and $\mathbf{u}^{(1)} = [1\ \ -1]^T$, but not a second solution. Our first guess for another one will be to assume something of the form $\mathbf{u}^{(2)}te^{2t}$ (keeping r the same), as we did for second-order ODEs when this challenge arose. In order for this to be a solution, it must satisfy $\mathbf{x}' = \mathbf{M}\mathbf{x}$, or alternatively the following:

$$2\mathbf{u}^{(2)}te^{2t} + \mathbf{u}^{(2)}e^{2t} = \begin{bmatrix} 1 & -1 \\ 1 & 3 \end{bmatrix}\mathbf{u}^{(2)}te^{2t} \tag{2.44}$$

However, this equation cannot be true unless $\mathbf{u}^{(2)} = 0$, since we have a term of e^{2t} on the left-hand side that does not show up anywhere on the right-hand side. This means that we have to assume instead that our second solution takes the form of $\mathbf{v}te^{2t} + \mathbf{w}e^{2t}$, so that we can properly balance terms. We then get the following:

$$2\mathbf{v}te^{2t} + (\mathbf{v} + 2\mathbf{w})e^{2t} = \mathbf{M}(\mathbf{v}te^{2t} + \mathbf{w}e^{2t}) = \begin{bmatrix} 1 & -1 \\ 1 & 3 \end{bmatrix}(\mathbf{v}te^{2t} + \mathbf{w}e^{2t}) \tag{2.45}$$

If we set the coefficients on both sides equal to each other, we get $2\mathbf{v} = \mathbf{M}\mathbf{v}$ and $(\mathbf{v}+2\mathbf{w}) = \mathbf{M}\mathbf{w}$. The first of these equations just states that $\mathbf{v}$ needs to be an eigenvector of $\mathbf{M}$. As we previously found $[1\ -1]^T$ to be such an eigenvector, we will use that for $\mathbf{v}$. For the second equation, we need that $(\mathbf{M} - 2\mathbf{I})\mathbf{w} = \mathbf{v} = [1\ -1]^T$. Solving this is only slightly different than solving an eigenvalue problem, as we will still get a relation between w_1 and w_2. Specifically, we get the following:

$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{2.46}$$

It therefore follows that if $w_1 = k$, $w_2 = -k - 1$. From this, we can obtain $\mathbf{w}$:

$$\mathbf{w} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} + k \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{2.47}$$

We therefore have everything we need to construct our second solution, which is the following:

$$\mathbf{x} = \mathbf{v}te^{2t} + \mathbf{w}e^{2t} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} te^{2t} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} e^{2t} + k \begin{bmatrix} 1 \\ -1 \end{bmatrix} e^{2t} \tag{2.48}$$

The last of these terms has the same form as what we got for the first solution and hence adds nothing new. Therefore, we can ignore it by assuming that $k = 0$. After this, taking a linear combination of the above solution with the first one will yield the general solution of the system.

What if we have eigenvalues with multiplicity more than 2? This can come up if we have a system of ODEs that is larger than just 2×2, for instance an eigenvalue of multiplicity 3 in a 3×3 system. In that case, you will just need extra powers of t to keep generating basis functions that are linearly independent. So, if your first basis function is some vector times e^{rt} for the repeated eigenvalue r, and your second is $\mathbf{v}te^{rt} + \mathbf{w}e^{rt}$ for vectors $\mathbf{v}$ and $\mathbf{w}$ (as we went through in the above example), then a hypothetical third basis vector would be $\mathbf{a}t^2e^{rt} + \mathbf{b}te^{rt} + \mathbf{c}e^{rt}$ for vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, a fourth one would also include a term with t^3e^{rt}, and so on.

So far, we've just looked at homogeneous systems. What happens when we include a forcing term? In that case, it's actually quite easy to get solutions, so long as the matrix $\mathbf{M}$ corresponding to the system is diagonalizable. (This is equivalent to it being of full rank, or in other words, having n linearly independent eigenvectors if it is an $n \times n$ matrix.) So, suppose we have the problem $\mathbf{x}' = \mathbf{M}\mathbf{x} + \mathbf{g}(t)$, where $\mathbf{g}$ is a vector of time-dependent functions (i.e. our forcing terms for each ODE). Now, let $\mathbf{T}$ be a matrix whose columns are the eigenvectors of $\mathbf{M}$, and let $\mathbf{x} = \mathbf{T}\mathbf{y}$ for some vector $\mathbf{y}$. We can substitute this into our equation $\mathbf{x}' = \mathbf{M}\mathbf{x} + \mathbf{g}(t)$ to get the following:

$$\mathbf{T}\mathbf{y}' = \mathbf{M}\mathbf{T}\mathbf{y} + \mathbf{g}(t) \tag{2.49}$$

The derivative of $\mathbf{T}\mathbf{y}$ is just $\mathbf{T}\mathbf{y}'$, as all of the entries in $\mathbf{T}$ are constants. When we left-multiply everything in this equation by the inverse of $\mathbf{T}$ (namely $\mathbf{T}^{-1}$) to isolate $\mathbf{y}'$, we get the following:

$$\mathbf{y}' = (\mathbf{T}^{-1}\mathbf{M}\mathbf{T})\mathbf{y} + \mathbf{T}^{-1}\mathbf{g}(t) \tag{2.50}$$

However, because of our choice for $\mathbf{T}$, the matrix $\mathbf{T}^{-1}\mathbf{M}\mathbf{T}$ is a diagonal matrix whose entries are the eigenvalues of $\mathbf{M}$, in the same columns as the corresponding eigenvectors of $\mathbf{M}$ are in $\mathbf{T}$. Additionally, $\mathbf{T}^{-1}\mathbf{g}$ is just another vector of time-dependent functions (which we will call $\mathbf{h}$). Because the matrix $\mathbf{T}^{-1}\mathbf{M}\mathbf{T}$ is diagonal and $\mathbf{T}^{-1}\mathbf{g}$ contains no variables other than t, we now have an uncoupled system, in which we can solve each equation separately. Specifically, we will get the following, for $i = 1, \ldots, n$ and r_i the eigenvalues of $\mathbf{M}$:

$$y_i'(t) = r_i y_i + h_i(t) \tag{2.51}$$

This means that we can solve for each y_i using techniques that we already know. After constructing our vector $\mathbf{y}$, we just need to left-multiply it by $\mathbf{T}$ to get the solution $\mathbf{x}$, since we defined $\mathbf{x} = \mathbf{T}\mathbf{y}$ earlier.

Chapter 3

Fixed points and bifurcations

3.1 November 12: Introduction to numerical integration, nonlinear dynamics, and mathematical modelling

Throughout this course, we have focused mostly on differential equations (or systems thereof) which are easy to solve by hand. Hence, you've mainly seen solutions that are composed of exponentials, trigonometric functions (which are just complex exponentials), and polynomials (often in the form of Taylor series). This is because the theory behind differential equations first arose during the 1800s, when calculations were typically done by hand. However, at the same time, physicists and other scientists were coming up with differential equations that described natural phenomena, of which most were (and are) quite difficult to find an analytical solution for. Bessel's equation, for instance, takes the following form:

$$x^2y'' + xy' + (x^2 - \alpha^2)y = 0 \tag{3.1}$$

This is still a linear differential equation, and its singular point at $x = 0$ is regular, so we at least have the tools to solve it by hand if we wanted to. However, we can't get a closed-form solution, and are instead left with an infinite polynomial series (here α is as defined above):

$$J_\alpha(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n! \cdot \Gamma(n + \alpha + 1)} \left(\frac{x}{2}\right)^{2n+\alpha} \tag{3.2}$$

Despite the fact that Bessel's equation looks relatively "nice" compared to other ODEs that we could come up with, finding the solution by hand is very challenging, and even requires us to use the Gamma function. If we restricted ourselves to solving by hand, this would leave us with a dilemma: we can use ODEs to describe processes in physics, chemistry, biology, and other fields that are definitely worth knowing, but solving them by hand is difficult. This is actually part of what prompted the theory behind numerical integration: the Runge-Kutta method, which is still one of the most popular methods for numerically integrating ODEs today, was developed at the beginning of the 1900s (after more complicated differential equations had proliferated) by Carl Runge and Wilhelm Kutta.

Additionally, in this course, we have only briefly touched upon systems of differential equations, otherwise known as dynamical systems. This is for the same reasons: once you get beyond linear systems with constant coefficients, analytical solutions are hard or impossible to find. This is despite the fact that a great many problems arise from the need to understand how different things interact with each other, which necessitates solving differential equations that are coupled together in a system rather than being independent from one another. I will illustrate this with an example. Suppose a biologist is studying two species, of which one is a predator and one is its prey. The biologist won't know the specific functions that describe the population sizes of the prey ($N(t)$) and predators ($P(t)$), but might be able to observe their rates of change in the field. Therefore, to predict how the populations of the two species will change over time, the biologist might construct two ODEs (one for each species) using the information regarding the rates of change. This process of building differential equations (and dynamical systems) that describe things in real life based on their observed or predicted rates of change is called mathematical modelling.

Let's see if we can put together this predator-prey system ourselves. We'll start with the ODE for the prey, which is $\frac{dN}{dt}$. We know that with plentiful food and adequate living space, a population will tend to grow exponentially. The reason for this is because we can model the growth of the population by assuming that over one unit of time, each individual in the population will produce r offspring, for some constant r. This is the birth rate of the population, which can be determined from field data. If we made the admittedly unrealistic assumption that our population could grow forever, we would arrive at this ODE:

$$\frac{dN}{dt} = rN \tag{3.3}$$

However, just as organisms are born, they can also die. We assumed that there was a predator that would eat the prey, so let's consider how that would affect the rate of change of the prey population. Obviously, if there are more predators, the odds of a prey organism encountering one (and being eaten by it) becomes greater. Likewise, if the number of predators is constant, then they will have a greater chance of running into prey if there is more prey. This leads us to a term that scales with both predator and prey population sizes that describes how often the prey get eaten by predators. We can add it to our ODE describing the prey population:

$$\frac{dN}{dt} = rN - \alpha NP \tag{3.4}$$

What about the predators? What rates govern how they are born and die? For this, we will consider the life cycle of a predator. They gain energy by eating prey, and use that energy to carry out all of their vital processes (including reproduction). So, we can assume that how many predators are born depends not only on how many predators there already are, but how many prey organisms there are as well. The predator death rate is simpler: since we have assumed that the predators don't have any predators themselves, we can just say that some constant proportion of the predator population will die over a given unit of time. This brings us to the following ODE for the predators:

$$\frac{dP}{dt} = \beta NP - mP \tag{3.5}$$

Combining the two into a 2×2 system gives us the following:

$$\begin{cases} \frac{dN}{dt} = rN - \alpha NP \\ \frac{dP}{dt} = \beta NP - mP \end{cases} \tag{3.6}$$

Since the populations of the two species can be directly measured in the field, we can also specify an initial condition:

$$\begin{cases} N(t=0) = N_0 \\ P(t=0) = P_0 \end{cases} \tag{3.7}$$

This is called the Lotka-Volterra model, which was independently derived by Alfred Lotka and Vito Volterra in the 1920s. One of its most famous applications is to explain the dynamics of lynx and hare populations in northern Canada; the data collected by the Hudson's Bay Company on the number of lynx and hare pelts its trappers obtained fits the model very well, even over the very long timescale over which these statistics were tracked.

Notice that both the predator and prey ODEs depend on both N and P, and that this dependence is nonlinear (due to the NP term) in both equations. This doesn't look like anything that we know how to solve. Indeed, it's all but impossible to do this analytically. (I recommend plugging the input $x' = x - xy$, $y' = xy - y$ into Wolfram Alpha to see what it gives as an output.) Therefore, we need to find out the properties of this function some other way.

Here's another example of how we can derive a mathematical model from first principles. Suppose that a chemical reaction is going on inside a cell, in which an enzyme is converting a substrate molecule into a reaction product. The substrate can freely bind to and disengage from the enzyme, but the enzyme-catalyzed reaction converting the substrate to the product is one-way. This means that we have the following reaction schematic:

$$E + S \rightleftharpoons ES \to E + P \tag{3.8}$$

There are four different interacting objects in our system, namely the enzyme (E), the substrate (S), the enzyme-substrate complex (ES), and the product (P). Therefore, in order to model it, we'll need four ODEs, which leads to a larger system than anything we've seen before. Fortunately, we have everything we need to do so in the above reaction schematic. We can assume that the enzyme and substrate bind together at a rate k_1 and unbind at a rate k_{-1}, and that once the complex is formed, the reaction proceeds at a rate k_2. Additionally, as with predators and prey, we can assume that how often the enzyme and substrate encounter each other (and hence bind to form the complex) is proportional to how much of both of them there is within the cell. Using this information, we can write a system of ODEs that characterizes the chemical reaction, or specifically how the concentrations of the different molecules change over time. Using the notation $e = [E]$, $s = [S]$, $c = [ES]$ ("c" for "complex"), and $p = [P]$, we have the following:

$$\begin{cases} \frac{de}{dt} = -k_1 se + k_{-1} c + k_2 c \\ \frac{ds}{dt} = -k_1 se + k_{-1} c \\ \frac{dc}{dt} = k_1 se - k_{-1} c - k_2 c \\ \frac{dp}{dt} = k_2 c \end{cases} \tag{3.9}$$

If we are initializing the model at the beginning of the reaction, we will have the following initial conditions:

$$\begin{cases} e(t=0) = e_0 \\ s(t=0) = s_0 \\ c(t=0) = 0 \\ p(t=0) = 0 \end{cases} \tag{3.10}$$

One thing that jumps out from the system of 4 ODEs is the fact that all of the terms in $\frac{de}{dt}$ are the opposites of the terms in $\frac{dc}{dt}$. In other words, we can see that $\frac{de}{dt} + \frac{dc}{dt} = 0$, or equivalently $e + c = e_0 + 0 = e_0$ given our initial conditions. This makes sense biologically, as the total amount of enzyme present (including enzyme that is both bound and unbound to the substrate) will remain constant throughout the reaction. This means that $e + c$ is a conserved quantity, which is important. It means that we can simplify our system of ODEs by taking $e(t) = e_0 - c(t)$, reducing the dimensionality by 1 and giving us the following:

$$\begin{cases} \frac{ds}{dt} = -k_1 s(e_0 - c) + k_{-1} c \\ \frac{dc}{dt} = k_1 s(e_0 - c) - (k_{-1} + k_2) c \\ \frac{dp}{dt} = k_2 c \end{cases} \tag{3.11}$$

This is still a nonlinear system, and none of the tools we have learned so far can be used to solve it. (It's still doable, though, so long as you make a few assumptions beforehand.)

As you can see from what we did above, it's pretty easy to generate a dynamical system if you already know something like a reaction diagram, which shows in graphical form all of the processes where some molecules are produced from other molecules. This means that even for large, complicated biochemical processes involving many different molecules (like the one we saw during lecture), we can construct a dynamical system that describes the process just by following the arrows in the reaction diagram. This concept can be generalized to other fields in which you might want to model a large network of interacting objects. To create a "reaction diagram", you can just draw each object in the system as its own box, and then draw arrows between boxes that interact with each other. Note that some of these reactions may be positive (i.e. the presence of Object A causes more of Object B to be produced), and some may be negative (i.e. Object A instead causes Object B to be consumed). Additionally, the functions that describe the interactions between the different boxes may be different than just the linear terms we've mostly seen in this course. For example, in the enzyme kinetics discussed above, some terms were linear and some terms were not, depending on the order of the reaction that each term represented. Other common terms that you'll see in mathematical models include logistic growth:

$$\frac{dx}{dt} = rx\left(1 - \frac{x}{K}\right) \tag{3.12}$$

as well as the saturation function:

$$\frac{dx}{dt} = \frac{x}{k + x} \tag{3.13}$$

and a generalized version of the saturation function, which is referred to in biology as the Hill function:

$$\frac{dx}{dt} = \frac{x^n}{k + x^n} \tag{3.14}$$

Of course, when you are creating a model, the most important part is making sure that your mathematical terms match what is observed in real life. This means that any arrow that you draw between boxes (and the functional form that you choose to represent the boxes' interaction) needs to have some rationale based on the problem at hand.

3.2 November 15: Steady states and phase portraits

Previously, I introduced the concept of nonlinear dynamical systems, which will generally be very hard or (usually) impossible to solve by hand. That means that we need to gain information about the state variables in such a system using other means. This week, I'll talk to you about many of the things that you can do with nonlinear dynamical systems without even having to integrate them at all.

One of the most important features of a dynamical system is its steady states. These are points in the system for which all rates of change are zero. If a dynamical system is at a steady state, then it will stay there. (An exception to this is if we are dealing with stochastic differential equations, and some random perturbation knocks the system off of the steady state, but that is beyond the scope of this course.) Other words for a steady state that you may encounter include "fixed point" and "equilibrium point". Mathematically, a steady state of a dynamical system is defined as anywhere at which the derivatives that make up the system are all zero, as at such a point none of the system's state variables can change. In other words, suppose we have the following system:

$$\begin{cases} \frac{dx_1}{dt} = f_1(t, x_1, x_2, \ldots, x_n) \\ \frac{dx_2}{dt} = f_2(t, x_1, x_2, \ldots, x_n) \\ \cdots \\ \frac{dx_n}{dt} = f_n(t, x_1, x_2, \ldots, x_n) \end{cases} \tag{3.15}$$

A fixed point $\mathbf{x}^* = [x_1^* \; x_2^* \; \ldots \; x_n^*]^T$ is any point in which $f_i = 0 \; \forall i$. This means that no change to any of the state variables $x_1, \; x_2, \; \ldots, \; x_n$ will occur. In practice, this is easiest to get when none of the functions f_i have any dependence on t, i.e. all differential equations in the system are autonomous.

To find where one of the state variables in the system (say x_1) has a rate of change of zero, we just need to set f_1 to zero and solve for values of $x_1, \; x_2, \; \ldots, \; x_n$ that will make

this so. If these values also make the rates of change of all of the other state variables zero, then they form a fixed point of the dynamical system (as defined above). Here's an example of this in action. Suppose that we have the following dynamical system:

$$\begin{cases} \frac{dx}{dt} = x^2 - 2y \\ \frac{dy}{dt} = 3x - xy \end{cases} \tag{3.16}$$

Let's start by setting $\frac{dy}{dt} = 0$. This occurs when either $x = 0$ or $y = 3$, since we get the relation $3x = xy$. Note that these are both lines in the (x, y)-plane rather than individual points. This will typically be the case when finding fixed points, since setting ODEs to zero will result in a relation between several variables (in the two-dimensional case, often a function of one variable with respect to the other). Any line in a 2×2 system on which one of the variables has a rate of change of zero is called a "nullcline"; the term "isocline" is also used, although this more properly refers to any line on which the slope of an ODE takes a specified constant value (not necessarily zero).

Next, we will take $\frac{dx}{dt} = 0$ to get the other condition for a fixed point. This results in the quadratic equation $y = \frac{1}{2}x^2$, which is the nullcline for x. Any point that is on the nullclines of both x and y will be a fixed point of the dynamical system. If you're finding fixed points in a 2×2 system, then you can plot the nullclines on a graph of y versus x to help visualize them and where they intersect. Any such plot where we graph two variables against each other rather than one of them against time is called the "phase plane", and an analogue in higher dimensions is "phase space". Once we have drawn the nullclines, we can also indicate regions of (x, y)-space where x is increasing or decreasing, depending on which side of the x-nullcline it is on, and likewise for y. This will allow us to get a rough approximation of our solutions without doing any integration at all. We'll see more things we can do with a phase plane later on.

In the case that we are working with, there will be three fixed points. This is because $y = \frac{1}{2}x^2$ is a parabola that opens upward, so it will intersect with the line $y = 3$ in two places, and it will intersect $x = 0$ at the point $(x^*, y^*) = (0, 0)$. To find the other two, we just need to find any x such that $\frac{1}{2}x^2 = 3$, which is true when $x = \pm\sqrt{6}$. Therefore, our other two fixed points are $(x^*, y^*) = (\sqrt{6}, 3)$ and $(x^*, y^*) = (-\sqrt{6}, 3)$.

It's possible for a system to have arbitrarily many fixed points. Take, for example, this relatively simple system:

$$\begin{cases} \frac{dx}{dt} = x \\ \frac{dy}{dt} = -2x \end{cases} \tag{3.17}$$

Here, the nullclines for x and y are the same, specifically $x = 0$, so the point $(x^*, y^*) = (0, y)$ for any real y will be a fixed point. This can be seen by integrating $\frac{dx}{dt}$ by hand. We get $x(t) = Ce^t$, but if we assume that x starts at 0 (for an initial condition), we get $C = 0$ and hence $x = 0$ because e^t cannot be 0 for finite t. Therefore, x is unchanging if it starts at 0, and since $\frac{dy}{dt}$ depends only on x, y will also never change no matter what initial condition is picked for it.

It's also possible for a system to have no fixed points. Consider this 3×3 system:

$$\begin{cases} \frac{dx}{dt} = x - z \\ \frac{dy}{dt} = y - x \\ \frac{dz}{dt} = z - x - 2 \end{cases} \tag{3.18}$$

If we take $\frac{dx}{dt} = 0$ and $\frac{dz}{dt} = 0$, we get two parallel planes in (x, y, z)-space. These can never intersect, so we can never get all three rates of change to be zero. There are places where two of the three state variables will be fixed, since the plane $y = x$ that we find by setting $\frac{dy}{dt} = 0$ intersects both of the other planes; the line $x = y = z$ is one of these.

What if our dynamical system is large and highly nonlinear, and the equations that we get when setting each rate of change equal to zero are hard to solve? In that case, we can use root-finding methods to obtain accurate approximations. You may have seen Newton's method in the past, most likely as a way to find roots of functions of a single variable. If you haven't, then it is defined as follows. Suppose that we want to find a root of a function $f(t)$ whose derivative exists. Then, starting at some initial guess t_0, we can apply the following formula to get a (usually) better guess t_1:

$$t_1 = t_0 - \frac{f(t_0)}{f'(t_0)} \tag{3.19}$$

This is a recurrence relation, so can be repeated additional times to come closer to the root and eventually get a very good approximation of it. We would like to have a way to generalize this to functions of multiple variables, as if we have this, we can find the roots of the functions $\frac{dx_1}{dt} = f_1(t, x_1, x_2, \ldots, x_n)$, $\frac{dx_2}{dt} = f_2(t, x_1, x_2, \ldots, x_n)$, and so forth. Luckily, such a way exists. The multi-dimensional analogue of the derivative is the Jacobian. Suppose that we have an autonomous dynamical system of the following form:

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1, x_2, \ldots, x_n) \\ \frac{dx_2}{dt} = f_2(x_1, x_2, \ldots, x_n) \\ \cdots \\ \frac{dx_n}{dt} = f_n(x_1, x_2, \ldots, x_n) \end{cases} \tag{3.20}$$

We can treat $f_1, f_2, \ldots, f_n$ as the entries of a vector-valued function, which we will call $\mathbf{F}$. The Jacobian of this function looks like this:

$$J_{\mathbf{F}}(x_1, x_2, \ldots, x_n) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{3.21}$$

If this is evaluated at a single point in $(x_1, \ldots, x_n)$-space, then it becomes a matrix full of constants. If this matrix is invertible, then we can use Newton's method, albeit a modified version in which we left-multiply F by the inverse of the Jacobian, $J_{\mathbf{F}}^{-1}$, instead of multiplying a univariate function f by $\frac{1}{f'}$. Thus, if $\mathbf{x}$ represents our vector of guesses for the fixed point of the system and $\mathbf{x}_0$ our initial guess, we can use the following formula to iterate towards a fixed point:

$$\mathbf{x}_1 = \mathbf{x}_0 - J_{\mathbf{F}}^{-1}(\mathbf{x}_0)\mathbf{F}(\mathbf{x}_0) \tag{3.22}$$

Note that taking the inverse of a matrix is unbelievably computationally expensive, i.e. it takes an extremely long time for a computer to do it (especially for big matrices). Therefore, faster methods have been developed to solve linear equations such as the one above. However, that is a topic for a different day.

Now that we know our fixed points, what can we say about them? Since fixed points represent the zeros of the vector-valued function with entries being the rates of change of our state variables, it would make sense that the rates of change of the state variables in the area around a fixed point would be very small. This is correct, but more important is which direction those small rates of change are in, particularly if they point towards or away from the fixed point. In the former case, any solution that starts sufficiently close to the fixed point will be drawn in towards it as time increases. In the latter case, solutions will be pushed away from it as time increases, although they would be drawn towards it if time is run backwards to $-\infty$. One example of this is the two differential equations $\frac{dx}{dt} = x$ and $\frac{dx}{dt} = -x$. Both of these have a single fixed point, namely $x = 0$. For $\frac{dx}{dt} = -x$, this is an "attractive" fixed point, which can be seen as the analytical solution to that ODE ($x(t) = e^{-t}$) tends towards the fixed point $x^* = 0$ as t increases. However, for $\frac{dx}{dt} = x$, $x^* = 0$ is instead a "repelling" fixed point as its solution ($x(t) = e^t$) moves away from the fixed point as time increases.

How do we tell whether a fixed point is attractive or repelling? It depends on the slopes of the functions f_1, f_2, and so on, specifically their slopes evaluated at the fixed point. This is because a fixed point $\mathbf{x}^*$ of the system is a place where the vector-valued function $\mathbf{F}$ (using the notation that we introduced above) is zero for all entries in the vector. Suppose we go a slight distance away from the fixed point, say to some point $\mathbf{x}^* + \mathbf{h}$ for some vector $\mathbf{h}$ whose entries are very small. The value of $\mathbf{F}$ is just the rates of change of all of our state variables. Therefore, if $\mathbf{x}^*$ is an attractive fixed point, then we want the slope of $\mathbf{F}$ to be in the opposite direction as our perturbation $\mathbf{h}$, so the action of $\mathbf{F}$ pushes us back into the fixed point. Similarly, for a repelling fixed point, we want $\mathbf{F}$ to act in the the same direction as the perturbation, so perturbing a solution away from the fixed point causes it to move even further away. I'll illustrate this with a one-dimensional example, for simplicity. Suppose we have this very straightforward ODE:

$$\frac{dx}{dt} = f(x) = x \tag{3.23}$$

Here, the only fixed point is $x^* = 0$. Therefore, we will be interested in the behaviour of $f(x)$ at some point $x = 0 + h = h$. If we take h to be positive, then $f(h)$ will also be positive, since it will just be h. Likewise, if we take h to be negative, then $f(h)$ will also be negative. In this way, perturbing some hypothetical solution $x(t)$ a small distance away from the equilibrium at 0 causes $\frac{dx}{dt} = f(x)$ to push it further away from the equilibrium, so $x^* = 0$ is a repelling fixed point. This can be confirmed by integrating this DE by hand and noticing that solutions tend to move away from 0 as time increases. Similarly, if we took $\frac{dx}{dt} = -x$, we would get exactly the opposite result. There, 0 is an attractive fixed point, consistent with the observed behaviour of exponential decay. For a general one-dimensional ODE $\frac{dx}{dt} = f(x)$,

the sign of the derivative $f'(x)$ evaluated at the fixed point $x = x^*$ determines whether the fixed point is attractive or repelling. If $f'(x^*)$ is negative, the fixed point is attractive. If it's positive, then the fixed point is repelling. This can be verified for the cases mentioned above, as $\frac{d}{dx}(x) = 1 > 0$ and $\frac{d}{dx}(-x) = -1 < 0$ regardless of which point they are evaluated at.

3.3 November 17: Stability of fixed points in higher dimensions, including a modelling example

Previously, we looked at how to determine if a fixed point for a one-dimensional system (i.e. a single differential equation) was attractive or repelling. Today, we'll do the same for larger systems. Before we do, I have a final note on the one-dimensional case. We showed that for a system $x' = f(x)$ and a fixed point x^*, x^* is attractive, or "stable", if $f'(x^*) < 0$, and it is repelling, or "unstable", if $f'(x^*) > 0$. What if $f'(x^*) = 0$? In that case, we don't have enough information to tell whether or not the fixed point is stable or unstable, and we will need to look at the slope of f around the fixed point rather than just at it. It could even be "semi-stable", in which solution trajectories that start on one side of the fixed point flow towards it and those that start on the other side flow away from it. To visualize this, consider the following ODE:

$$\frac{dx}{dt} = x^2 \tag{3.24}$$

This has a fixed point at $x^* = 0$, but we can see that $f'(x) = x$, which also takes the value of 0 at $x = x^*$. However, $f'(x)$ takes negative values for negative values of x, and positive values for positive values of x. Based on what we said about the sign of f', we would thus expect a solution $x(t)$ that started with a negative value of x to flow towards the fixed point, and a solution $x(t)$ that started with a positive value of x to flow away from the fixed point. This is, in fact, true. If you integrate the above ODE by hand, you will get the solution $x(t) = \frac{1}{C-t}$ for C a constant of integration. This solution is a hyperbola with its singularity at $t = C$ and the horizontal axis as an asymptote; it increases towards ∞ when it is positive and decreases towards $-\infty$ when it is negative.

As a sidenote, the process of evaluating the stability of a fixed point of a nonlinear dynamical system by evaluating the slopes of the functions making up the system is called "linearization". The reason for the name is linked to the reason why it works. Suppose we have a one-dimensional ODE, $x' = f(x)$, which has a fixed point x^*. Without loss of generality, we can assume that this fixed point is $x^* = 0$, since if it is some other value of x we can just do a coordinate transform to make it 0 (e.g. $u = x - 3$ for a fixed point $x^* = 3$). If this is a linear ODE, i.e. $f(x) = kx$ for some k, then determining the stability is easy, because we can integrate by hand and get either exponential growth or exponential decay. (In this case, the derivative f' is also just a constant, making it easy by this method as well.) However, for nonlinear f, integration by hand might be difficult. On the other hand, all we want to know is the behaviour of a solution in some neighbourhood of the fixed point, not over all possible values of t. Therefore, we can make a linear approximation to the function by using the Taylor series:

$$f(x) = x^* + f'(x^*)(x - x^*) + \mathcal{O}(x^2) \tag{3.25}$$

Note that if $x^* = 0$, we get a linear function in x with only one term, which can easily be integrated by hand. In general, this will be a pretty good approximation close to the fixed point, since the linear term of the Taylor expansion will dominate all of the others close to 0. However, the more nonlinear our function is, the more error will accumulate as we move away from the fixed point. If we are far away from the fixed point under consideration, the solution might do something that its linear approximation wouldn't, such as converge to a different fixed point.

So, let's dive in to higher-dimensional systems. Using the notation that we introduced on Monday, suppose that we have the following dynamical system:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}) \tag{3.26}$$

As with the one-dimensional case, we want to find out what the slope of $\mathbf{F}$ is at values very close to the fixed point, or in other words $\mathbf{F}(\mathbf{x}^* + \mathbf{h})$ for small $\mathbf{h}$. We will still do this by linearization, but in multiple dimensions, this is more complicated than just looking at the sign of the derivative of f, like we did in the one-dimensional case. Instead, we'll look at the Jacobian of our system (which we talked about earlier), since the linear term in the Taylor expansion for a vector-valued function involves the Jacobian:

$$\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x}^*) + J_{\mathbf{F}}(\mathbf{x}^*) \cdot (\mathbf{x} - \mathbf{x}^*) \tag{3.27}$$

Another way to look at this is that the Jacobian encompasses all of the partial derivatives of the functions in $\mathbf{F}$, and therefore the slopes of all of the functions in $\mathbf{F}$ with respect to all of the variables in the system. Note also that if our system isn't autonomous, then we might have some terms in the Jacobian with t, based on how the partial derivatives turn out. In that case, linearization requires some extra justifications, which I won't go into here.

So, we have that the linearization of our function $\mathbf{F}$ involves taking the Jacobian of $\mathbf{F}$ at the fixed point $\mathbf{x}^*$. How do we use this to evaluate the stability of $\mathbf{x}^*$? Well, instead of just taking the sign of f' in the one-dimensional case, we will take the signs of all of the eigenvalues of $J_{\mathbf{F}}$. More specifically, we will take the signs of all of the real parts of the eigenvalues of $J_{\mathbf{F}}$, This is because real-valued matrices can have complex eigenvalues (as we have seen), but the complex parts of these don't affect convergence to the fixed point $\mathbf{x}^*$, because we have been operating under the assumption that $\mathbf{x}^*$ is real and $\mathbf{F}$ is real-valued.

If all of the eigenvalues have negative real parts, then perturbing our solution by some slight amount in any direction in phase space away from the fixed point will cause the trajectory of the solution to fall back into the fixed point. This means that the fixed point will be stable, or a "sink". If at least one eigenvalue has a positive real part, then solutions will eventually escape along the eigendirection in phase space associated with that eigenvalue, making it unstable. However, there are two different ways that this can happen. If every eigenvalue has a positive real part, then solution trajectories will escape from the fixed point in any direction in phase space. Such a fixed point is called a "source"; note that if time is run backwards, sources become sinks and vice versa. If some eigenvalues have positive real parts and some have negative real parts, then solution trajectories may approach the

fixed point along an eigendirection associated with a negative eigenvalue, but then move away from it along an eigendirection associated with a positive eigenvalue. These kinds of fixed points are called "saddle points". To see why, picture some small object sitting in the middle of a saddle, perfectly balanced. If you perturb it forward or backward with no lateral motion, then it would theoretically roll back into the fixed point in the middle. However, if you perturbed it to the left or right, it would roll off. If instead the small object started out somewhere other than the very middle of the saddle, it would initially roll towards the fixed point at the middle, but its ultimate fate would be to fall off. Plotting a solution trajectory in phase space would reveal a similar pattern, with "forward or backward" corresponding to a stable eigendirection and "left or right" corresponding to an unstable one.

I will illustrate this with a few simple examples. Suppose we have the following system, of which the solution should be obvious:

$$\begin{cases} \frac{dx}{dt} = x \\ \frac{dy}{dt} = 2y \end{cases} \tag{3.28}$$

Here, we have one fixed point, which is the origin. This system is already linear, so finding the Jacobian of it is trivial. We get the following:

$$J_{\mathbf{F}} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \tag{3.29}$$

This is a diagonal matrix, so its eigenvalues are just the diagonal entries, namely $r = 1$ and $r = 2$. (If you really want to calculate the characteristic polynomial, it's $r^2 - 3r + 2$.) Both of these are positive, so we get that the origin is a source, which lines up with what we know about exponential functions. If, instead, we had the following dynamical system:

$$\begin{cases} \frac{dx}{dt} = -x \\ \frac{dy}{dt} = -2y \end{cases} \tag{3.30}$$

then we would get the origin to be a sink, which once again confirms what we can find analytically. What about a saddle point? Well, suppose we have the following dynamical system:

$$\begin{cases} \frac{dx}{dt} = x + y \\ \frac{dy}{dt} = x - y \end{cases} \tag{3.31}$$

This is a coupled linear system, and we can solve it analytically using the methods that we have already learned. It has the origin as its only fixed point, as that is the only place where the lines $x + y = 0$ and $x - y = 0$ cross. Taking the Jacobian of this system (which is the same as the coefficient matrix, as the system is linear), we get the following:

$$J_{\mathbf{F}} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.32}$$

The characteristic polynomial of this matrix is $r^2 - 2 = 0$, meaning that our eigenvalues will be $r = \pm\sqrt{2}$. From what we know about the theory behind linear systems, we can

say that our general solution will include a term with $e^{\sqrt{2}t}$ and a term with $e^{-\sqrt{2}t}$. As t increases, depending on the coefficients on the terms (which we can determine using the initial conditions), we could see trajectories in phase space (i.e. the (x, y)-plane) in which the solution (x, y) first appears to approach the origin, but then moves away from it. (A time series of $y(t)$ would show something similar.) This can be seen, of course, by the fact that one of the eigenvalues of the Jacobian is positive and the other is negative.

What if the eigenvalues are complex? If that is the case, then the behaviour of our solutions in phase space will involve rotation. (This is intuitive to see. Think of the behaviour of e^{kt} when k is real versus when k is complex or purely imaginary, and the fact that the sine and cosine functions in (x, y)-space are linked to rotation around a unit circle.) In this case, if the fixed point that we are evaluating the Jacobian at is a sink or a source, then solutions will travel towards or away from the fixed point (respectively) in a spiral manner in phase space. (Saddle points won't see much change in practice.) If the eigenvalues are purely imaginary, then solutions will neither converge to nor escape from the fixed point, instead rotating around it in a circle, at least theoretically. In practice, this is more accurate for linear systems, since for these the Jacobian is the same as the coordinate matrix that is used when solving by hand, as seen above. For nonlinear systems, the extra terms present in the Taylor expansion that get ignored during linearization may cause the eigenvalues to have nonzero real parts. We might still get cyclic patterns, but if we do, it's highly unlikely that they'll be circular.

Since we have just looked at linear systems so far, let's see an example of the linearization process for evaluating the stability of a fixed point of a nonlinear system. Suppose we have the following system, which we used last week as a predator-prey model:

$$\begin{cases} \frac{dN}{dt} = rN - \alpha NP \\ \frac{dP}{dt} = \beta NP - mP \end{cases} \tag{3.33}$$

We will assume that all constants in the model are non-negative, since the system becomes non-biological otherwise (e.g. "I eat you and then there's more of you"). The first step here is to find the fixed points. $\frac{dN}{dt} = 0$ when either $N = 0$ or $P = \frac{r}{\alpha}$, and $\frac{dP}{dt} = 0$ when either $P = 0$ or $N = \frac{m}{\beta}$. Therefore, there are two locations in phase space where an N-nullcline intersects with a P-nullcline, which are $(N^*, P^*) = (0, 0)$ and $(N^*, P^*) = (\frac{m}{\beta}, \frac{r}{\alpha})$. The first of these represents extinction of both species, and the second represents coexistence.

Now, let's evaluate the Jacobian at these points. If we take the partial derivatives of both of the functions making up the dynamical system, we get the following:

$$J_{\mathbf{F}} = \begin{bmatrix} r - \alpha P & -\alpha N \\ \beta P & \beta N - m \end{bmatrix} \tag{3.34}$$

At $(0, 0)$, most of the terms in the Jacobian reduce to zero, and we are left with the following:

$$J_{\mathbf{F}} = \begin{bmatrix} r & 0 \\ 0 & -m \end{bmatrix} \tag{3.35}$$

If we assume that both r and m are positive, then the origin is therefore a saddle point and hence unstable. In particular, the positive eigenvalue will be r; you can calculate the

corresponding eigenvector (and hence the eigendirection that solutions will escape from $(0,0)$ on) yourself. A biological interpretation of this is that so long as the two species aren't both completely extinct, they will continue to survive in the long term. What about the other fixed point, $(\frac{m}{\beta}, \frac{r}{\alpha})$? In this case, we get the following for our Jacobian:

$$J_{\mathbf{F}} = \begin{bmatrix} 0 & \frac{-\alpha m}{\beta} \\ \frac{r\beta}{\alpha} & 0 \end{bmatrix} \tag{3.36}$$

Note that we have zeros on the diagonal, and that the other two terms have opposite signs. That's a clue that we'll have purely imaginary eigenvalues, and indeed our characteristic polynomial is $r^2 + rm$, which has the roots $r = i\sqrt{rm}$. We had previously said that this might indicate periodic orbits of the fixed point, although as the system is nonlinear, more analysis is needed. In order to see whether or not these periodic solutions exist, since we can't readily integrate the functions in this system with respect to t, we will instead look at the movement of solutions within the phase plane, i.e. in (N, P)-space. Consider the following differential equation:

$$\frac{dP}{dN} = \frac{dP}{dt}\left(\frac{dN}{dt}\right)^{-1} = \frac{\beta NP - mP}{rN - \alpha NP} = \frac{P(\beta N - m)}{N(r - \alpha P)} \tag{3.37}$$

This describes the movement of P relative to N, and luckily for us it is separable. Separating and integrating yields the following:

$$\int \frac{r - \alpha P}{P}\, dP = \int \frac{\beta N - m}{N}\, dN \tag{3.38}$$

This evaluates to the following expression, after moving around some symbols and combining constants of integration:

$$r \ln P - \alpha P - \beta N + m \ln N = C \tag{3.39}$$

We have from before that $N = 0$ is an N-isocline and that $P = 0$ is a P-isocline. From this, we can conclude that if N starts off positive, then it can't become negative, and likewise for P. In other words, any solution that starts in the first quadrant of the (N, P)-plane will stay there. This is reassuring, since if it weren't true then this model wouldn't be very realistic. Additionally, we just derived that the expression $r \ln P - \alpha P - \beta N + m \ln N$ must always be equal to a constant, which is finite. This prevents N or P from blowing up to infinity, so long as both species exist. (Since $\ln N$ and $\ln P$ could be $-\infty$ if N or P is zero, we could get the prey going to ∞ if there are no predators present. If there is no prey, $\frac{dP}{dt}$ will be strictly negative for initial conditions in the first quadrant, so the predators cannot escape to ∞.) Based on these conclusions, we can confidently say that both the predator and prey populations will follow bounded, periodic solutions.

3.4 November 19: Introduction to bifurcations

Previously, we worked through finding the fixed points of Lotka-Volterra predator-prey model and their stability. This model was a bit different from the other dynamical systems that

you may have seen before, since it included some parameters. These are any constant in the model that is left unspecified (but is not a state variable or t). In the Lotka-Volterra model, these were r, α, β, and m. We made the assumption that all of these were positive, which caused us to make some conclusions about our fixed points, namely that $(N^*, P^*) = (0, 0)$ is a saddle point and that $(N^*, P^*) = (\frac{m}{\beta}, \frac{r}{\alpha})$ admits periodic orbits. However, in other dynamical systems, the parameters might not be defined so strictly. Another way of saying this is that there might be a fixed point for which the stability changes depending on the values of certain parameters in the system (even if we make an assumption that all parameters are positive or something similar). It's also possible for a dynamical system to have fixed points that only exist for certain ranges of a given parameter; if the parameter is outside these ranges, then $\mathbf{F}(\mathbf{x})$ evaluated at that "fixed" point might be nonzero. These occurrences, as well as other significant qualitative changes in the behaviour of a dynamical system when one of its parameters is changed, are called "bifurcations".

I will demonstrate with a simple example. Suppose we have the following one-dimensional ODE, for k a constant:

$$\frac{dx}{dt} = f(x) = k - x^2 \tag{3.40}$$

Let's try finding the fixed points of this DE. They occur when $x^2 = k$, which means that we will see different behaviour when $k > 0$, $k = 0$ and $k < 0$. For $k > 0$, the equation $x^2 = k$ has two real solutions, which means that there will be two fixed points, $x^* = \sqrt{k}$ and $x^* = -\sqrt{k}$. In order to evaluate the stability of these, we will take the derivative of $f(x)$, which is $f'(x) = -2x$. Evaluated at $\sqrt{k}$, this is a negative number (since $\sqrt{k}$ is positive), which means that the fixed point $x^* = \sqrt{k}$ is stable. Likewise, we get that the fixed point $x^* = -\sqrt{k}$ is unstable by using the same method.

On the other hand, if we let $k < 0$, then there are no fixed points in this system, because the values of x that make $\frac{dx}{dt} = 0$ will be purely imaginary. Taking $k = 0$ makes the system reduce to $\frac{dx}{dt} = -x^2$, which has just one fixed point at $x^* = 0$ rather than the two that appear when $k > 0$. In this case, $f'(x)$ evaluated at the single fixed point $x^* = 0$ is also zero, which means that we can't draw any conclusions about its stability from evaluating $f'(x)$. (We saw previously that $\frac{dx}{dt} = x^2$ has a semi-stable fixed point at $x^* = 0$, and $\frac{dx}{dt} = -x^2$ is the same, albeit with the areas in which the fixed point attracts or repels solutions being swapped.) Because our ODE $\frac{dx}{dt} = k - x^2$ has a major change to its fixed points at the parameter value $k = 0$, we say that a bifurcation happens at $k = 0$.

A good way to visualize the fixed points of a system and how they can change with different values of a certain parameter is by drawing a bifurcation plot. This is a graph with a parameter on the horizontal axis and the fixed point of some state variable on the vertical axis. In other words, given a dynamical system, we are plotting the value taken by one of the variables in the system at one of the system's fixed points as a function of a parameter. For the system we saw previously, we can plot x^* as a function of k. On the right-hand side of this bifurcation plot, where $k > 0$, we will get two lines showing the locations of our two fixed points. (By convention, on a bifurcation plot, stable fixed points are shown as solid lines, while unstable ones are shown as dashed lines.) Since the locations of these two fixed points are $x^* = \pm\sqrt{k}$, the graph looks like what we would get if we plotted the two equations $y(t) = \sqrt{t}$ and $y(t) = -\sqrt{t}$ as part of a standard time series. These two lines representing

the fixed points will collide with each other at $k = 0$, and on the left-hand side of the graph no lines representing fixed points will exist. This kind of bifurcation, in which a stable fixed point and an unstable fixed point come together and annihilate one another, is called a "fold bifurcation", as on a bifurcation plot it looks like the line representing the fixed point is being folded over. Fold bifurcations are something to watch out for in mathematical models, as they can indicate the possibility of drastic changes in the trajectory of a solution if the underlying parameter is altered in some way.

What are some other kinds of bifurcations? Let's look at another one-dimensional ODE, once again for k a constant:

$$\frac{dx}{dt} = f(x) = kx - x^2 \tag{3.41}$$

This one has fixed points when $kx = x^2$, or in other words when $x = k$ and $x = 0$. Taking the derivative of f results in $f'(x) = k - 2x$. When we evaluate the stability of $x^* = k$, we can notice that f' reduces to $-k$. This means that if $k > 0$, $x^* = k$ is stable, and if $k < 0$ it is unstable. Meanwhile, if we evaluate the stability of $x^* = 0$, we arrive at the opposite conclusion, because there f' reduces instead to k. Therefore, at $k = 0$, the stability of both of the fixed points changes, with $x^* = k$ going from unstable to stable and $x^* = 0$ going from stable to unstable. This also corresponds to the intersection of $x^* = 0$ and $x^* = k$ on a bifurcation plot. (Hence, at $k = 0$ only one fixed point will exist, and we won't be able to ascertain its stability by taking f' because f' will be zero.) However, unlike in the example we saw previously, the two lines on the bifurcation plot cross at an angle rather than colliding head-on, so we get a change in stability rather than them obliterating each other. This kind of bifurcation, in which two fixed points cross each other and change each other's stability, is called a "transcritical bifurcation".

Are there more kinds of bifurcations? Of course there are. Consider the following system, for k a constant:

$$\frac{dx}{dt} = kx - x^3 \tag{3.42}$$

This has fixed points whenever $x^3 = kx$, which happens when $x = 0$ and when $x = \pm\sqrt{k}$. Therefore, there can be a maximum of three fixed points for this dynamical system, but this only occurs when $k > 0$. For $k < 0$, $\pm\sqrt{k}$ will both be imaginary, so $x^* = 0$ will be the only fixed point. So far, we already know that something interesting happens at $k = 0$, namely the creation of two additional fixed points. What happens when we evaluate their stability? Well, $f'(x) = k - 3x^2$. When we test $x^* = 0$, this reduces to just k, so $x^* = 0$ is stable if $k < 0$ and unstable if $k > 0$. For $x^* = \pm\sqrt{k}$, f' evaluates to $k - 3k$, which is negative for the only values of k for which those two fixed points exist (i.e. $k > 0$). Therefore, at $k = 0$, a lot of different things happen. The existing fixed point in the system switches from being stable to unstable, but two new stable fixed points are created on either side of it. If you plot all of the fixed points on a bifurcation plot, the shape of all of the lines resembles a pitchfork, and hence this kind of bifurcation is called a "pitchfork bifurcation". Note that you can also get pitchfork bifurcations where a fixed point goes from unstable to stable and two new unstable fixed points are created on either side of it. For an example of this, look at $\frac{dx}{dt} = kx + x^3$. A pitchfork bifurcation which has one stable fixed point that branches into two stable and

one unstable ones is called "supercritical", whereas a pitchfork bifurcation which has one unstable fixed point that branches into two unstable and one stable fixed points is called "subcritical".

There's another kind of bifurcation that only occurs in systems of two dimensions or higher. This is called the Hopf bifurcation, and it is related to a dynamical system having periodic solutions. Before I begin explaining it, I would first like to point out that having a dynamical system with at least two state variables is a prerequisite for obtaining a periodic solution. You can see this by noting that for a one-dimensional ODE $\frac{dx}{dt} = f(x)$, the value of f' evaluated at any real fixed point will itself be a real number, but for a dynamical system of dimension 2 or higher, the eigenvalues of the Jacobian evaluated at a fixed point might be imaginary. A periodic solution to a dynamical system is also referred to as an "orbit", as that's what a drawing of it in a phase plane will resemble. If it serves as the limit for some solution trajectories as either time goes to ∞ or time goes to $-\infty$, then it is referred to as a "limit cycle" (stable and unstable, respectively). Periodic solutions that are solely composed of sines and cosines are not limit cycles, since a dynamical system with sines and cosines as its general solution will have each individual solution be a circle in phase space (and thus no solution curve will ever converge to another one). However, if you have a nonlinear dynamical system that admits periodic solutions, these will be limit cycles.

So, what is a Hopf bifurcation? Well, consider the following dynamical system, for k a parameter:

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -x + (k - x^2)y \end{cases} \tag{3.43}$$

This is what we get when we transform a second-order ODE called the Liénard equation (which is $x'' - (k - x^2)x' + x = 0$) into a system of first-order ODEs. The only fixed point for this system is the origin, as you can see by setting up the nullclines. The Jacobian of this system evaluated at the origin is as follows:

$$J_{\mathbf{F}} = \begin{bmatrix} 0 & 1 \\ -1 & (k - x^2) \end{bmatrix} \implies J_{\mathbf{F}}(0,0) = \begin{bmatrix} 0 & 1 \\ -1 & k \end{bmatrix} \tag{3.44}$$

The characteristic equation for this matrix is $r^2 - kr + 1 = 0$, so we get solutions of the following form:

$$r = \frac{k}{2} \pm \frac{\sqrt{k^2 - 4}}{2} = \frac{k}{2} \pm \frac{i\sqrt{4 - k^2}}{2} \tag{3.45}$$

When the absolute value of k is small, the eigenvalues will be complex conjugates. If k is negative, then the real part of both eigenvalues will be negative, so the origin will be a stable node (specifically a spiral because of the imaginary parts). However, when $k = 0$, the real part disappears and we are left with periodic orbits for our solutions. What about when $k > 0$? We would expect the origin to become an unstable node. However, this is only part of what actually happens. In addition to the origin switching from stable to unstable, a limit cycle appears around the origin. This new limit cycle is stable, which means that it soaks up any solution trajectory that starts somewhere near it. (In this case, as the origin is the only fixed point and it's unstable, any solution to our dynamical system that starts

anywhere other than the origin will converge to this new limit cycle as time goes to ∞.) This is the result of the Hopf bifurcation. The general criteria for a Hopf bifurcation are as follows. Suppose that we have a dynamical system including some parameter k, which has a fixed point $\mathbf{x}^*$. Suppose also that there is a value of k (which we will call k_0) for which all of the eigenvalues of the Jacobian evaluated at $\mathbf{x}^*$ have negative real parts, except for one pair of eigenvalues that are purely imaginary. In other words, suppose that this pair of eigenvalues is of the form $g(k) \pm ih(k)$ in general, but for $k = k_0$ we get $g(k_0) = 0$ and $h(k_0) \neq 0$. Now, suppose further that $\frac{dg}{dk}(k = k_0) > 0$. Then, there exists some k_1 such that the dynamical system in question has a periodic orbit surrounding $\mathbf{x}^*$ for $k_0 < k < k_1$. (If $\frac{dg}{dk}(k = k_0) < 0$, then we instead get an orbit for $k_1 < k < k_0$, as the values of k for which $g(k) > 0$ and $g(k) < 0$ will be reversed.)

In practice, the existence of Hopf bifurcations means that an unstable equilibrium in 2×2 systems or higher will often be surrounded by a limit cycle. Here's an example of a mathematical model in which they occur. Consider the following dynamical system:

$$\begin{cases} \frac{dx}{dt} = -x + ay + x^2 y \\ \frac{dy}{dt} = b - ay - x^2 y \end{cases} \tag{3.46}$$

This is the Sel'kov model of glycolysis, which is a simplified version of a more complicated enzyme kinetic model (similar to the one that you saw earlier) after a few biological assumptions were made about some reactions being very fast compared to others. This model has a fixed point at $(x^*, y^*) = (b, \frac{b}{a+b^2})$. The Jacobian is as follows:

$$J_{\mathbf{F}} = \begin{bmatrix} -1 + 2xy & a + x^2 \\ -2xy & -a - x^2 \end{bmatrix} \implies J_{\mathbf{F}}(x^*, y^*) = \begin{bmatrix} -1 + \frac{2b^2}{a+b^2} & a + b^2 \\ \frac{-2b^2}{a+b^2} & -a - b^2 \end{bmatrix} \tag{3.47}$$

Calculating the eigenvalues of this matrix is quite heavy on the algebra, so I'll omit it here. (Of course, you could also calculate them numerically.) They will take the form of a conjugate pair, with the real part as follows:

$$r = g(a, b) \pm ih(a, b) \implies g(a, b) = \frac{a + a^2 - b^2 + 2ab^2 + b^4}{-2(a + b^2)} \tag{3.48}$$

If we want this to be zero, we will need a and b such that everything in the numerator drops. Once again, this is algebraically rather involved, but you will eventually end up with the following values for b as a function of a:

$$b(a) = \sqrt{\frac{1}{2}\left(1 - 2a \pm \sqrt{1 - 8a}\right)} \tag{3.49}$$

Note that there are two values here, in keeping with the fact that $g(a, b)$ has higher powers of both a and b. This means that there will be two Hopf bifurcations. Indeed, if you plot the solutions for the Sel'kov model for a fixed value of a and varying values of b, you will see your solutions tend towards a stable node to start, followed by a stable limit cycle, then back to a stable node. While the limit cycle exists, there will be an unstable node inside it (at the same location that the stable node would be otherwise). However, this unstable node will be essentially impossible to hit if you're doing numerical simulations, because the nature

of floating-point arithmetic and finite step sizes means that you will always be making slight perturbations away from any unstable fixed point in a system that you're simulating.

3.5 November 22: More about periodic orbits and the stability of fixed points

Last week, we saw some examples of periodic orbits of dynamical systems. These are any solutions for which the trajectory of all of the state variables follows a closed, deterministic path throughout phase space, which doesn't have to be defined by sine and cosine. You may recall that for the Lotka-Volterra predator-prey system, we showed that there were periodic orbits. Here is the Lotka-Volterra system, for reference:

$$\begin{cases} \frac{dN}{dt} = rN - \alpha NP \\ \frac{dP}{dt} = \beta NP - mP \end{cases} \tag{3.50}$$

To determine that periodic orbits exist for this system, we first found a fixed point for which the eigenvalues are purely imaginary, which is $(x^*, y^*) = (\frac{m}{\beta}, \frac{r}{\alpha})$, and noted that the nullclines of $N = 0$ and $P = 0$ made it impossible for a solution that started in the first quadrant (i.e. with $N, P \geq 0$) to leave it. Then, we took the derivative $\frac{dP}{dN}$ to represent the motion of this system in the phase plane, then showed that integrating $\frac{dP}{dN}$ resulted in a finite conserved quantity that can be expressed as a function of N and P. This meant that solutions for N and P in the first quadrant for which $0 < N, P < \infty$ had to stay within those bounds, and specifically follow a trajectory in the phase plane defined by what we got when we integrated $\frac{dP}{dN}$. (Note that in Cartesian coordinates, $x^2 + y^2 = 1$ defines the unit circle, and $m \ln N + r \ln P - \beta N - \alpha P = C$ also defines a closed curve in the (N, P)-plane.)

We also talked about how periodic orbits can arise due to a Hopf bifurcation. Specifically, a Hopf bifurcation can occur when all of the eigenvalues of the Jacobian evaluated at a fixed point of a dynamical system have negative real parts, except for two that are purely imaginary conjugates of each other. If some parameter in the system is changed so that the two imaginary conjugate eigenvalues shift to having positive real parts, then the fixed point in question switches from stable to unstable, but a stable limit cycle is created surrounding the fixed point.

There are a few more results that may be useful for finding periodic orbits. To illustrate these, suppose we have the following autonomous dynamical system:

$$\begin{cases} \frac{dx}{dt} = f(x, y) \\ \frac{dy}{dt} = g(x, y) \end{cases} \tag{3.51}$$

We choose this system to be autonomous because having time dependence can potentially throw a solution off of a limit cycle, just like with a fixed point. Anyway, one important result concerns where orbits and limit cycles can and cannot appear in this system's phase plane (i.e. (x, y)-space). Any closed trajectory (in other words, an orbit) of a two-dimensional dynamical system like the one above must enclose at least one fixed point in the phase plane. Furthermore, if the orbit only encloses one fixed point, then that fixed point cannot

be a saddle. (We saw this for Hopf bifurcations, where a limit cycle must enclose a node.) As a matter of fact, any orbit in the phase plane must enclose an odd number of fixed points, $2n+1$ of them for n a non-negative integer, of which n are saddles and $n+1$ are either sinks (stable nodes) or sources (unstable nodes). These findings come from a field called index theory, which is a bit beyond the scope of this course. A handy way to remember them (which also comes from index theory) is to say that in the phase plane, the index of a source or sink is $+1$, the index of a saddle point is -1, the index of a periodic orbit is also $+1$, and the sum of any curve in the phase plane (regardless of whether it is an orbit) is the sum of the indices of any fixed points that it encloses. This method allows us to demonstrate that a given curve in the phase plane is not an orbit (if it has an index of anything other than $+1$), as well as to rule out orbits entirely in some cases (if it is impossible to draw a closed curve with index $+1$).

Another result is an application of Green's theorem (from Calculus 3) to dynamical systems. Suppose that we have the two-dimensional dynamical system mentioned above, and both $f(x,y)$ and $g(x,y)$ have continuous first partial derivatives. Suppose also that we have some simply connected region R in the phase plane. If the function $\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y}$ does not change sign anywhere in R, then our system $x' = f(x,y),\ y' = g(x,y)$ has no closed orbits in R. As a matter of fact, this is a specific case of a more general result. If we pick any function $u(x,y)$ that is continuously differentiable, such that the function $\frac{\partial}{\partial x}(uf) + \frac{\partial}{\partial y}(ug)$ does not change sign in R, then there are no closed orbits of our system in R. This general case is called Dulac's criterion. It can be used to rule out periodic orbits in regions of phase space, but one downside is that it requires coming up with a suitable function $u(x,y)$.

We now have some tools for determining that there are no closed orbits in some region of the (x,y)-plane for our dynamical system $x' = f(x,y),\ y' = g(x,y)$. What if we want to prove that an orbit exists in some region? That one's a little tougher. However, we do have one result that we can use. Suppose we have some region R in the phase plane that is bounded, contains its boundary, does not contain any fixed points of the system $x' = f(x,y),\ y' = g(x,y)$, and on which $f(x,y)$ and $g(x,y)$ are both continuously differentiable. If there is some solution trajectory that remains in R for all values of t greater than some fixed value t_0, then R contains a periodic orbit of the system. (Note that R cannot be simply connected, since the periodic orbit in question needs to enclose a fixed point of the system, which cannot itself be in R.) This result is called the Poincaré-Bendixson Theorem.

So, we now have a few results on the topic of orbits that we can use. What about fixed points? Is it possible to determine the stability of a fixed point without having to go through all the trouble of finding the eigenvalues of the Jacobian? (This is a very legitimate question for large systems, since computing anything involving large matrices can get quite intense.) It turns out that there is. One method is by finding what is called a Lyapunov function. Suppose that we have an autonomous dynamical system of the following form:

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1, \ldots, x_n) \\ \cdots \\ \frac{dx_n}{dt} = f_n(x_1, \ldots, x_n) \end{cases} \tag{3.52}$$

Suppose also that this dynamical system has an isolated fixed point at the origin. (Mathematically, this means that there is some neighbourhood of the fixed point at the origin which

no other fixed points are in.) If it has one that is not at the origin, then we can move it there by using a coordinate transform. Now, consider a function $V(x_1, \ldots, x_n)$ that is "positive definite". This means that V is equal to zero at the origin and is positive everywhere else. Consider also the time derivative of V:

$$\frac{dV}{dt} = \frac{\partial V}{\partial x_1}\frac{dx_1}{dt} + \ldots + \frac{\partial V}{\partial x_n}\frac{dx_n}{dt} \tag{3.53}$$

If the time derivative of V is "negative definite", i.e. it is zero at the origin and negative everywhere else, then the origin is a stable fixed point of the system under consideration, and we call V a "Lyapunov function". More specifically, the origin will be what we call "asymptotically stable", which means that there exists some $\delta > 0$ such that all solution trajectories that start a distance less than δ away from the origin will have their distance towards the origin go to zero in the limit case as $t \to \infty$. (This is more or less equivalent to the origin being a sink.) We get a stronger result if V is also radially unbounded. "Radially unbounded" means that $||\mathbf{x}|| \to \infty \implies V(\mathbf{x}) \to \infty$, for some norm $||\cdot||$ and $\mathbf{x}$ being the vector with entries $x_1, \ldots, x_n$. We can take $||\cdot||$ to be any norm, so the 1-norm $x_1 + \ldots + x_n$ or the 2-norm (which is the Euclidean distance) will suffice. In this case, then the origin is "globally asymptotically stable", which means that δ can be chosen to be any finite positive number (and hence the origin attracts solutions that start anywhere in phase space). If $\frac{dV}{dt}$ is only negative definite in some bounded region surrounding the origin, then we can only prove that solutions starting in that region tend to the origin. On the other hand, if $\frac{dV}{dt}$ is positive definite, then the origin is unstable. The theory behind Lyapunov functions can produce very strong results, particularly for large systems when other methods might be infeasible. However, as with Dulac's criterion, it requires coming up with a suitable function.

Let's see a simple example of this. Suppose we have the following system:

$$\begin{cases} \frac{dx}{dt} = -x \\ \frac{dy}{dt} = -y \end{cases} \tag{3.54}$$

We know based on past experience that there is one fixed point for this system, namely the origin, and that it is stable. Can we prove this using a Lyapunov function? Let's try $V(x, y) = x^2 + y^2$. This is clearly positive definite, and it is also radially unbounded, since any combination of x and y that goes to infinity also makes V go to infinity. We will now calculate the time derivative of V:

$$\frac{dV}{dt} = \frac{\partial V}{\partial x}\frac{dx}{dt} + \frac{\partial V}{\partial y}\frac{dy}{dt} = -2x^2 - 2y^2 \tag{3.55}$$

This is negative definite, so we can conclude that the origin is a globally asymptotically stable fixed point for this dynamical system.

How about a more challenging example? Suppose we have the following system:

$$\begin{cases} \frac{dx}{dt} = -x + y^2 \\ \frac{dy}{dt} = 3x^2 - 2y \end{cases} \tag{3.56}$$

This system has a fixed point at the origin, but it also has another one at $(x^*, y^*) = \left(\left(\frac{2}{3}\right)^{2/3}, \left(\frac{2}{3}\right)^{1/3}\right)$. Therefore, the origin cannot be globally asymptotically stable, because it

cannot attract solutions that start at the other fixed point. However, we can still try to find a Lyapunov function. Consider the following positive definite, radially unbounded function:

$$V(x,y) = \frac{x^2}{2} + \frac{y^2}{4} \tag{3.57}$$

The time derivative of this is as follows:

$$\frac{dV}{dt} = -x^2 + xy^2 - y^2 + \frac{3}{2}yx^2 = -x^2(1 - \frac{3}{2}y) - y^2(1-x) \tag{3.58}$$

This is negative definite when $x < 1$ and $y < \frac{2}{3}$, so we at least know that solutions that start within those bounds will tend towards the origin. (The actual region of phase space in which solutions tend towards the origin is actually much bigger, but this is as far as Lyapunov functions will take us.)

Another tool in our arsenal for evaluating the stability of fixed points is the Routh-Hurwitz criterion. If we already have evaluated the Jacobian at a fixed point, then we will be able to find the characteristic polynomial. However, characteristic polynomials of order 3 or 4 might be hard to find roots for, and of course any polynomial of order at least 5 cannot be solved algebraically. This means that we need some other way to assess the stability of fixed points in systems that large. Anyway, suppose that we have a characteristic equation as follows:

$$a_n r^n + a_{n-1} r^{n-1} + \ldots + a_1 r^1 + a_0 = 0 \tag{3.59}$$

We won't be able to algebraically determine the roots of this polynomial if it's large enough, but we will be able to determine if they have negative real parts based solely on the coefficients of the polynomial. This will be done by using the Routh-Hurwitz stability criterion, and to use this criterion we first need to construct a table using the coefficients called the "Routh array". If our characteristic polynomial is of degree n, then our table will have n rows, which we will denote r^n, r^{n-1}, and so on down to r^1. The first two rows of the table will be constructed out of alternating coefficients in the characteristic equation, with a_n being the first entry:

$$\begin{array}{c|cccc} r^n & a_n & a_{n-2} & a_{n-4} & \ldots \\ r^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \ldots \end{array} \tag{3.60}$$

If we run out of coefficients for a particular row, we just let everything to the right of the last coefficient in that row be zero. For instance, if n is even, then the first row will end with a_0, so there won't be any coefficent that we can put directly below a_0 in the second row (so we let that entry in the second row be zero).

To construct the rest of the rows in the table, we will use a recursive formula. If we call the entries of the third row b_1, b_2, b_3, and so on, then the formulae for them are as follows:

$$b_1 = \frac{-1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{bmatrix}, \; b_2 = \frac{-1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{bmatrix}, \; \ldots \tag{3.61}$$

If we extend down to the fourth row, the entries (which we can call c_1, c_2, c_3, etc.) will follow a similar pattern:

$$c_1 = \frac{-1}{b_1} \det \begin{bmatrix} a_{n-1} & a_{n-3} \\ b_1 & b_2 \end{bmatrix},\ c_2 = \frac{-1}{b_1} \det \begin{bmatrix} a_{n-1} & a_{n-5} \\ b_1 & b_3 \end{bmatrix},\ \ldots \tag{3.62}$$

Eventually, the Routh array will look like this, with q denoting the last entry obtained from this process:

$$\begin{array}{c|cccc} r^n & a_n & a_{n-2} & a_{n-4} & \cdots \\ r^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \cdots \\ r^{n-2} & b_1 & b_2 & b_3 & \cdots \\ r^{n-3} & c_1 & c_2 & c_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ r^0 & q & 0 & 0 & 0 \end{array} \tag{3.63}$$

The entries in any given row (besides the first two) will depend on the entries in the two rows directly above it. Additionally, when calculating any given entry, the left-hand column of the determinant in the formula for that entry will include the first entries in each of the two rows directly above the row that is being created, and the right-hand column of the determinant will gradually move right in terms of which entries it contains. The determinant will always be divided by the negative of the first entry in the row directly above the one currently being created. Mathematically, if we denote the Routh array as a matrix $\mathbf{M}$, then we get the following formula for $m_{i,j}$, for $i > 2$:

$$m_{i,j} = \frac{-1}{m_{(i-1),1}} \det \begin{bmatrix} m_{(i-2),1} & m_{(i-2),(j+1)} \\ m_{(i-1),1} & m_{(i-1),(j+1)} \end{bmatrix} \tag{3.64}$$

Now that we have our Routh array, we can use the actual Routh-Hurwitz criterion. The number of times that the sign of an entry in the first column is different than the sign of the entry directly above it is the number of roots that the characteristic polynomial has with positive real parts. Therefore, if the entries in the first column of the Routh array are either all positive or all negative, then the fixed point in question is stable. A simplified version of this criterion for second-order polynomials $r^2 + a_1 r + a_0$ is that the polynomial's roots will all have negative real parts if and only if a_1 and a_0 are both positive. For third-degree polynomials $r^3 + a_2 r^2 + a_1 r + a_0$, we need a_2 and a_0 to both be positive and $a_2 a_1 > a_0$.

Chapter 4

Numerical integration

4.1 November 24: Floating point arithmetic, Euler's method, and error bounds

Previously, we have learned about dynamical systems that may be hard or impossible to solve by hand. This is generally because they are nonlinear. Despite this, these kinds of dynamical systems make up the vast majority that you will encounter in real life, and so finding solutions to them is important. Therefore, we can use various numerical methods to approximate the solutions to the dynamical systems in question. Numerical integration is just one part of the field of numerical analysis, which also includes things like methods for approximating the eigenvalues of a matrix (which I will touch upon later on in this course).

In general, numerical integration of systems of differential equations relies on starting from a known quantity (i.e. the initial condition of the system), then using the rates of change of each state variable to approximate what a solution will look like afterwards. This process can be iterated over and over again to build a curve that approximates the analytical solution. One very simple way to illustrate this is with Euler's method. Suppose we have the following very simple one-dimensional ODE with an initial condition:

$$\frac{dx}{dt} = f(t, x),\ x(t = 0) = x_0 \tag{4.1}$$

We can express $\frac{dx}{dt}$ as a limit, using the form that you learned in Calculus 1. Using this, the ODE becomes the following:

$$\lim_{h \to 0} \frac{x(t + h) - x(t)}{h} = f(t, x) \tag{4.2}$$

However, we will not take this limit, and instead assume that h is a very small number. We can then do some algebra to get the following:

$$x(t + h) = x(t) + hf(t, x) \tag{4.3}$$

Now, we have a formula in which we can use $x(t)$, the value of our solution at the current time, and $f(t, x)$, the derivative evaluated at the current time, to find $x(t + h)$, the value of the solution at some time in the future. This is a recurrence relation, so we'll need something

to start with. An obvious choice would be x_0, since that was provided for us in the problem. We can use that to get x_h, then x_{2h}, x_{3h}, and so forth. In higher dimensions, the formula still holds, although our inputs and outputs will be vector-valued:

$$\begin{cases} \frac{dx_1}{dt} = f_1(t, x_1, \ldots, x_n) \\ \cdots \\ \frac{dx_n}{dt} = f_n(t, x_1, \ldots, x_n) \end{cases} \implies \begin{cases} x_1(t+h) = x_1(t) + hf_1(t, x_1, \ldots, x_n) \\ \cdots \\ x_n(t+h) = x_n(t) + hf_n(t, x_1, \ldots, x_n) \end{cases} \tag{4.4}$$

Now that we have this tool for approximating solutions, let's see if it works. We will test it out on an ODE that we know the solution to:

$$\frac{dx}{dt} = x^2, \; x(t=0) = -1 \tag{4.5}$$

We can solve this analytically to get $x(t) = \frac{-1}{t+1}$, which has a simple functional form for us to compare our answers with. Now, let's do a few steps of Euler's method. We'll start with $h = 0.1$ as our step size, to keep things tractable for now. We get the following:

$$\begin{aligned} &x(0) = -1 \\ &x(0.1) = -1 + 0.1 \cdot (-1)^2 = -0.9 \\ &x(0.2) = -0.9 + 0.1 \cdot (-0.9)^2 = -0.819 \\ &x(0.3) = -0.819 + 0.1 \cdot (-0.819)^2 \approx -0.752 \end{aligned} \tag{4.6}$$

Now, let's compare this to the actual values of our solution, to see how well we did. Here are the actual values of $x(t)$ at the points that we specified:

$$\begin{aligned} &x(0) = -1 \\ &x(0.1) = \tfrac{-1}{0.1+1} \approx -0.909 \\ &x(0.2) = \tfrac{-1}{0.2+1} \approx -0.833 \\ &x(0.3) = \tfrac{-1}{0.3+1} \approx -0.769 \end{aligned} \tag{4.7}$$

It's not terrible; we do capture the overall trend (increasing towards zero at a decreasing rate) as well as one digit of accuracy following the decimal point. The reason why we don't get closer estimates is mainly because we used a relatively large step size. We can see this by considering the Taylor expansion of $x(t)$ around some point t_0:

$$x(t) = x(t_0) + x'(t_0)(t - t_0) + \frac{x''(t_0)}{2}(t - t_0)^2 + \mathcal{O}(t^3) \tag{4.8}$$

Now, take $t = t_1$ for some time $t_1 > t_0$:

$$x(t_1) = x(t_0) + x'(t_0)(t_1 - t_0) + \frac{x''(t_0)}{2}(t_1 - t_0)^2 + \mathcal{O}((t_1 - t_0)^3) \tag{4.9}$$

We can specify $h = t_1 - t_0$, and we also know that $x'(t_0) = f(t_0)$. Therefore, the first two terms of the Taylor series turn into one step of Euler's method, and we get the following:

$$x(t_0 + h) = x(t_0) + hf(t_0) + \frac{x''(t_0)}{2}h^2 + \mathcal{O}(h^3) \tag{4.10}$$

Note that the t^2, t^3, etc. terms in the Taylor series will turn into h^2, h^3, etc., since the dependence is now on $(t_1 - t_0) = h$. We ignore everything from the third term onward when we perform a step of Euler's method, but those terms are still necessary parts of the solution (since the Taylor expansion is an infinite sum). Therefore, the difference between the real value of x at $t = t_0 + h$ and our simulated value (which we will call x_1, by analogue with $x(t_0) = x_0$) is the following:

$$\text{LTE} = x(t_0 + h) - x_1 = \frac{x''(t_0)}{2}h^2 + \mathcal{O}(h^3) \tag{4.11}$$

This is specifically the error that accumulates by taking one step with Euler's method, which we call the "local truncation error" (hence "LTE" above) because it results from truncation of the Taylor series. If we assume that $h < 1$, which is more or less always the case, then we get $h^2 > h^n \ \forall n > 2$. This allows us to combine the later terms in the Taylor series, and say that the local truncation error is on the order of h^2. (This holds so long as the third derivative x''' is bounded, since if it isn't, then we can't assume that the later terms in the Taylor series are less prominent than the h^2 term.) What about the overall error present in the approximation? In other words, how different will the curve that we simulate be from the actual solution? Well, if the beginning of our simulations is the point t_0 that we already specified, and we run the simulations until some other time t_{final}, then the total number of steps is $\frac{1}{h}(t_{\text{final}} - t_0)$. Therefore, if there is $\mathcal{O}(h^2)$ error in every step, then the total amount of error (the "global truncation error") will be the following:

$$\text{GTE} = \frac{t_{\text{final}} - t_0}{h}\frac{x''(t_0)}{2}h^2 = \mathcal{O}(h) \tag{4.12}$$

This is, as we can see, proportional to h. Therefore, the smaller we take h, the less truncation error (local and global) we get. Since the global truncation error is proportional to the first power of h, i.e. h^1, we say that Euler's method is a "first-order method". (If the global truncation error depended on h^2, it would be a second-order method, and so on.) What if we instead increased the step size, for instance taking $h = 1$? If we simulate the same ODE as before, we will get the following:

$$\begin{aligned} x(0) &= -1 \\ x(1) &= -1 + 1 \cdot (-1)^2 = 0 \\ x(2) &= 0 + 0 = 0 \end{aligned} \tag{4.13}$$

Having too big of a step size can make the method break down completely. In practice, your step size usually shouldn't be anywhere near $h = 1$; I mostly use $h = 10^{-3}$ myself if I don't need extra precision.

However, even though we showed that taking lower values for h leads to lower truncation error, that doesn't mean that we always need to choose as low a value for h as possible. There are a couple important reasons for this. The first one is that smaller h means more steps that need to be computed, which in turn means that your numerical integration will take more time. (One way to balance this is to integrate your dynamical system with two different values of h, one being the value that you are using and the other being a value one order of magnitude greater. If the difference between the two solutions produced is

acceptably small, then it's probably fine to use the larger value of h to save computation time.)

The other reason that lower h is not always better also has to do with the number of computations performed. Computers only have a finite amount of memory, and base-10 numbers are stored in a computer's memory in the form of base-2 approximations with a finite number of digits. This means that there are only a finite number of computer-representable numbers, and hence in most cases, the computer representation of a decimal number will not be the same as the form that we would write out by hand. Because of this, mathematical operations done on a computer will typically lose a small amount of accuracy, due to having to round off the operation's output to something which is representable by the computer. This is called "round-off error". An example of this is as follows. Consider the decimal number 0.1, or $(0.1)_{10}$. When converted into binary, it has an infinite number of decimal places:

$$(0.1)_{10} = (0.00011001100110011\ldots)_2 \tag{4.14}$$

This means that representing it on a computer (with finite precision as opposed to infinite precision) must necessarily eliminate all of its digits after a certain point, resulting in error. A corollary of this is that two numbers that differ by less than the computer's precision will be represented as the same number. For example, suppose we have a very primitive computer that can only handle five binary digits after the decimal point (i.e. up to 2^{-5}). This computer's representation of $(0.1)_{10}$ would be $(0.00011)_2$, but that would be the same as the representation of (for instance) $(0.1 + 2^{-6})_{10}$. In this way, two operations on our primitive computer that have expected outputs of $(0.1)_{10}$ and $(0.1 + 2^{-6})_{10}$ would in practice be indistinguishable. If h is large, or at least fairly large, the total amount of round-off error will be at least an order of magnitude smaller than the truncation error. However, if h is sufficiently small, round-off errors will build up and cause substantial total error if many computational steps are performed. Therefore, an ideal value for h would be not too large and not too small.

Previously, we found that the global truncation error of Euler's method was proportional to h. Do methods exist that have greater accuracy for a given step size? Yes, in fact there are several. One of the most commonly used ones is the Runge-Kutta fourth-order method, or RK4. Since this is a fourth-order method, its global truncation error will be $\mathcal{O}(h^4)$, and by extension its local truncation error will be $\mathcal{O}(h^5)$. This is much more accurate than Euler's method, which is first order and therefore has a global truncation error of $\mathcal{O}(h)$.

So, how does the fourth-order Runge-Kutta method work? The main concept behind it is that we will predict the next step in our numerical solution, $x(t + h)$, by using a weighted average of the predicted slopes of x in the interval $[t, t + h]$. It's more accurate for the same reasons that the trapezoidal rule is more accurate than the various rectangular approximation methods for calculating integrals, or why Simpson's method is more accurate than the trapezoidal rule. RK4 takes a weighted average of four different slopes in this interval. This means that the recurrence relation for getting the next step of a solution in RK4 has four terms with h in them. The specific formulation of RK4 is as follows, for an ODE $x' = f(t, x)$, an initial condition $x(t = 0) = x_0$, and the assumption that each step advances time t by h units:

$$\begin{aligned} x_{n+1} &= x_n + h\left(\tfrac{1}{6}k_1 + \tfrac{1}{3}k_2 + \tfrac{1}{3}k_3 + \tfrac{1}{6}k_4\right) \\ k_1 &= f\left(t_n, x_n\right) \\ k_2 &= f\left(t_n + \tfrac{h}{2}, x_n + k_1\tfrac{h}{2}\right) \\ k_3 &= f\left(t_n + \tfrac{h}{2}, x_n + k_2\tfrac{h}{2}\right) \\ k_4 &= f\left(t_n + h, x_n + k_3 h\right) \end{aligned} \tag{4.15}$$

Note that k_1 is linear in terms of h, k_2 involves taking h times k_1 and is thus quadratic in h, and likewise k_3 and k_4 are cubic and quartic in h, respectively. This is what makes the local truncation error on the order of h^5 (and hence the global truncation error on the order of h^4 once we multiply by $\frac{1}{h}(t-t_0)$). Let's iterate this a few times with the differential equation we looked at earlier ($x' = x^2$, $x(0) = -1$) and see if it does better than Euler's method. I'll omit the algebra that takes place in terms of calculating the steps, and just show you the results, to ten decimal places this time:

$$\begin{aligned} x(0) &= -1 \\ x(0.1) &\approx -0.9090911863 \\ x(0.2) &\approx -0.8333337288 \\ x(0.3) &\approx -0.7692312058 \end{aligned} \tag{4.16}$$

In this case (and in general), RK4 does indeed do better than Euler's method, and by a considerable margin. The actual solution values are (to ten decimal places) -0.9090909091, -0.8333333333, and -0.7692307692, which means that we get five digits of accuracy even with a relatively large step size of $h = 0.1$.

4.2 November 26: Multistep methods, stiff equations and stability

In the previous lecture, we looked at the fourth-order Runge-Kutta method, which is probably the most widely-used numerical integration scheme in the majority of cases. However, there are some cases where other methods may be desired. For instance, certain ODEs may have solutions that change very rapidly in time, such that accurately approximating a solution may be impossible without taking an extremely small step size. These are called "stiff" ODEs, and methods such as Euler's method may have trouble solving them. For stiff ODEs, we need to use methods that can handle these sharp changes in the solution; the ability of a numerical integration method to do this is referred to as its "stability".

For an illustrative example, consider the ODE $x' = -100x$ with initial condition $x(0) = 1$. This has an analytical solution of $x(t) = e^{-100t}$, and it should therefore decay to 0 quite quickly. What happens when we attempt to use Euler's method on it, with a step size of 0.1?

$$\begin{aligned} x(0) &= 1 \\ x(0.1) &= 1 + 0.1(-100 \cdot 1) = -9 \\ x(0.2) &= -9 + 0.1(-100 \cdot -9) = 81 \\ x(0.3) &= 81 + 0.1(-100 \cdot 81) = -729 \end{aligned} \tag{4.17}$$

Instead of monotonically converging to 0, our simulated solution diverges while oscillating about the horizontal axis, which is the opposite behaviour of what we wanted. It turns out that Euler's method is not particularly stable when stiff equations are concerned. One way to see this is the concept of "A-stability". A numerical integration scheme is A-stable if it successfully captures the fact that all ODEs of the form $x' = kx$, for k a constant with negative real part, tend to zero as $t \to \infty$. A related concept is that of the "stability region" of a numerical integration scheme. For a numerical scheme that takes x_{n+1} to be a function of x_n, integrating a test function of the form used in the definition of A-stability (namely $x' = kx$) means that the method of producing x_{n+1} from x_n will depend on the step size h and the constant k. As a matter of fact, for functions of this type, it will invariably depend on their product hk. Therefore, we can say that $x_{n+1} = u(hk)x_n$ for some function u, with the specific function depending on the numerical scheme used. It can clearly be seen that if $|u(hk)| < 1$, then $|x_{n+1}| < |x_n|$, and the simulated solution x will go to zero. The stability region for a numerical scheme is the set of complex numbers z such that $|u(z)| < 1$, with the implication that for a specified test function $x' = kx$ and step size h, we can check if the numerical scheme defined by $u(z) = u(hk)$ produces a solution that converges to zero. It follows that an A-stable numerical integration scheme is one for which the stability region includes the entire left half of the complex plane (i.e. all complex numbers with negative real parts), as that means that a simulation of $x' = kx$ will converge to zero for any k with negative real part, regardless of the step size h.

Let's see an example of this. We have previously attempted to solve a stiff equation using Euler's method, so we'll find the stability region of Euler's method first. The recurrence relation defining Euler's method is $x_{n+1} = x_n + hf(x_n)$, so for $f(x) = kx$, we have $x_{n+1} = x_n + hkx_n = x_n(hk + 1)$. It follows that the function used for finding the stability region is $u(hk) = hk + 1$, or $u(z) = z + 1$. Therefore, the actual stability region for Euler's method is $\{z \in \mathbb{C} : |z + 1| < 1\}$. This does not encompass all complex numbers with negative real parts, so Euler's method is not A-stable. We can also see that for $h = 0.1$ and $k = -100$, we get $hk = -10$ and $|hk + 1| = 9 > 1$, confirming what we saw earlier that Euler's method with a step size of $h = 0.1$ cannot successfully integrate the ODE $x' = -100x$. (In fact, the values taken by x during our simulations went up by a factor of 9 with every step, as the stability region calculations show.)

How about the fourth-order Runge-Kutta method? Expanding out the various coefficients that make up the method give us the following recurrence relation for x_{n+1} as a function of x_n:

$$u_{\mathrm{RK4}}(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \tag{4.18}$$

Note that this agrees with the Taylor expansion up to the fourth-order term, which is necessary for RK4 to be a fourth-order numerical integration scheme. The region of stability for RK4 is the area in which $|u_{\mathrm{RK4}}(z)| < 1$. Because $u_{\mathrm{RK4}}(z)$ is a fourth-degree polynomial function of z, it would be hard to graph this by hand (although you could do so using mathematical software). However, the most important thing is that the stability region does not cover all complex numbers with negative real parts. This means that RK4 is also not A-stable, and therefore that you will need very small step sizes to accurately capture the dynamics of very stiff systems using RK4. For the example that we used before with

$z = hk = 0.1 \cdot -100 = -10$, we get the following:

$$|u_{\text{RK4}}(-10)| = 1 - 10 + \frac{100}{2} - \frac{1000}{6} + \frac{10000}{24} = 291 > 1 \tag{4.19}$$

This one surprisingly does even worse than Euler's method, although it would be more manageable if we decreased the step size. (Remember that if $z = hk < 1$, then $z^n < 1$ for $n > 1$.) What if we wanted a method that was A-stable, or in other words that it would integrate problems of this type well regardless of the choices of h and k? This can actually be done, and it can be done relatively easily. The numerical integration schemes that we would use to satisfy such a constraint are all what we call "implicit methods", so named because when evaluating x_{n+1}, they do so using an implicit formula that includes x_{n+1} and x_n instead of providing x_{n+1} as an explicit function of x_n. One extremely straightforward implicit method is the implicit Euler method, also called the backward Euler method. This is nearly identical to the (forward) Euler method, but with one very important distinction. Remember that the forward Euler method defines x_{n+1} as being $x_{n+1} = x_n + hf(x_n)$, or $x_{n+1} = x_n + hf(t_n, x_n)$ for a time-dependent ODE $x' = f(t, x)$. The implicit formula for the backward Euler method is as follows:

$$x_{n+1} = x_n + hf(t_{n+1}, x(t_{n+1})) = x_n + hf(t_{n+1}, x_{n+1}) \tag{4.20}$$

The derivation of this comes from the concept of integrating $\frac{dx}{dt}$ to get x. If we take a definite integral of $\frac{dx}{dt} = f(t, x)$ from time t_n to time $t_{n+1} = t_n + h$, then we can evaluate it to get the following:

$$\int_{t_n}^{t_{n+1}} f(t, x(t))\ dt = x(t_{n+1}) - x(t_n) = x_{n+1} - x_n \tag{4.21}$$

However, we can also use a very simple Riemann sum (just one rectangle) to approximate the integral above. Taking the value of the function f at the left-hand bound of the interval would just result in the forward Euler method, so we'll use the value on the right-hand bound. Since the length of the interval that we are integrating over is h, we get the following approximation:

$$\int_{t_n}^{t_{n+1}} f(t, x(t))\ dt \approx hf(t_{n+1}, x(t_{n+1})) = hf(t_{n+1}, x_{n+1}) \tag{4.22}$$

Putting these together yields the formula for the backward Euler method, as described above. As this is an implicit formula, we will need to use the specific form of f to isolate x_{n+1}. Obviously, this may be challenging if f is complicated. Is the backward Euler method A-stable? Let's check the function defining its stability region. If we assume that f takes the form $f(x) = kx$, we get the following:

$$x_{n+1} = x_n + hf(x_{n+1}) \implies x_{n+1}(1 - hk) = x_n \implies u(z) = (1 - z)^{-1} \tag{4.23}$$

We can see that $|u(z)| < 1$ when $1 < |1 - z|$. This is true for all z with negative real parts, and even most z with positive real parts. Therefore, the backward Euler method is A-stable, although it may produce other inaccuracies depending on the step size chosen. However, we do have better options available to us. Since Euler's method is based on the

rectangle rule for approximating an integral, maybe a better method than the rectangle rule would produce a better numerical integration scheme. How about the trapezoid rule? If we perform the same steps as we did to derive the backward Euler method, but approximate the integral of $f(t, x(t))$ using the trapezoid rule instead of the rectangle rule, we get the following formula relating x_{n+1} and x_n:

$$x_{n+1} = x_n + \frac{h}{2}\left(f(t_n, x_n) + f(t_{n+1}, x_{n+1})\right) \tag{4.24}$$

This is the trapezoidal method for integrating ODEs, and it is also an implicit method. Note that we're averaging the values of f at t_n and t_{n+1}, hence the factor of $\frac{1}{2}$ (in addition to the step size h). This makes the trapezoidal method a "multistep method", since the equation $x_{n+1} - x_n$ that defines how much our solution will change by depends on the value of x at multiple different locations. So, how does this method do when evaluating stiff problems? We will calculate the region of stability for it to find out. If we assume that $f(x) = kx$, our formula for the trapezoidal method can be rewritten as the following:

$$x_{n+1} = x_n + \frac{h}{2}(kx_n + kx_{n+1}) \tag{4.25}$$

Solving for x_{n+1} yields the following:

$$x_{n+1} = \frac{1 + \frac{1}{2}hk}{1 - \frac{1}{2}hk}x_n \implies u(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \tag{4.26}$$

In order for $|u(z)|$ to be less than 1, we need that $|1 + \frac{1}{2}z| < |1 - \frac{1}{2}z|$. However, this is true whenever z is closer to -1 than to 1, or in other words the entire left half of the complex plane. From this, we get the result that the trapezoidal method is A-stable. We also get the stronger condition that for any ODE $x' = kx$, the simulated solution of that ODE using the trapezoidal method will converge to zero if and only if the actual solution does.

So, we now have a method that has been shown to work very well for stiff dynamical systems. The trapezoidal method is second-order; its local truncation error can be found to be $\mathcal{O}(h^3)$, and therefore its global truncation error is $\mathcal{O}(h^2)$. Can we improve upon this to get better accuracy (like we improved on the forward Euler method to get RK4), while keeping A-stability? The answer to this is actually that we can't. Explicit multistep methods cannot be A-stable, and implicit multistep methods that are third-order or higher can also never be A-stable. This result was proved by Germund Dahlquist in 1963 and is known as the second Dahlquist barrier. (This means that you have actually seen a result proved in the last 50 years in this class!)

Now that we know a good way to approach stiff dynamical systems, where can we expect to encounter them in real life? One important class of dynamical systems that may be stiff is those that contain a separation of timescales, when some processes that make up the system happen orders of magnitude more quickly than others. Here is a simple example of this, for ε a very small constant:

$$\begin{cases} \frac{dx}{dt} = f(x, y) \\ \varepsilon\frac{dy}{dt} = g(x, y) \end{cases} \tag{4.27}$$

Some chemical reaction networks contain ODEs that are on a faster timescale than the rest of the system like this, and population dynamics in which some species have much faster life cycles than others can also be modelled in this way. Additionally, most dynamical systems used in neuroscience are stiff. This is because they model (among other things) the electrical potential found in a neuron, which undergoes a large spike when the neuron fires. This, however, is a topic for next week.

Chapter 5

Mathematical modelling

5.1 November 29: Hodgkin-Huxley and FitzHugh-Nagumo models; constructing a model from first principles

Now that you've seen some of the theory behind differential equations, I will spend the last few lectures telling you about some examples of how they are used in real life. This should also give you some insight on how mathematical models are created; one of the purposes of this particular lecture is to show how one can build a model from first principles. One of the first, and most important, examples of a mathematical model is the Hodgkin-Huxley model of electrical activity in a neuron. This was created in 1952 by Alan Hodgkin and Andrew Huxley, and was fitted to real experimental data that they obtained by measuring the voltage along the giant axon of a squid. (The axon is giant, not the squid. Action potentials in the neuron that this particular axon is part of cause the squid to contract some of its muscles to expel water from its body, which propels the squid forward very rapidly and can allow it to escape potentially harmful situations.)

Since this is an electrical model, most of the theory behind it is based on electrical circuits. In particular, most of the terms in the Hodgkin-Huxley model consist of voltages and currents. For a brief explanation of how these are related, consider a capacitor in a circuit, which stores charge when a voltage is applied to it. Specifically, if a constant C represents the capacitance (the relative ability of the capacitor to store charge), then the charge Q stored by the capacitor can be expressed as a function of the applied voltage V:

$$Q(t) = C \cdot V(t) \tag{5.1}$$

The derivative of Q is the current intensity (or just "current") across the capacitor. To get this, we can take the derivative of both sides, which gives an expression in terms of $\frac{dV}{dt}$:

$$I(t) = C \cdot \frac{dV}{dt} \tag{5.2}$$

Indeed, within the context of the Hodgkin-Huxley model, the lipid bilayer that makes up most of the cell membrane of the squid giant axon is treated as a capacitor. Thus, if V_m represents the axon's "membrane potential" (the difference in electric potential between the

inside and outside of the axon), the current flowing across the lipid bilayer I_c can be written as the following:

$$I_c = C_m \frac{dV_m}{dt} \tag{5.3}$$

However, there are other ways in which ions can move in and out of the cell, which of course have implications for the electrical current. Two of the most important ions in animal cells are sodium and potassium. This is because animal cells maintain a roughly constant electrical potential by exporting sodium ions and importing potassium ions, causing both sodium and potassium ions to usually have a substantial gradient between the inside of the cell and the outside. For a given ion i, the "reversal potential" V_i is the membrane potential at which there is no net flow of the ion into or out of the cell. This means that ions contribute to the electrical current across the cell membrane depending on how much the actual membrane potential differs from each ion's reversal potential. Additionally, the transport of ions across a cell membrane is carried out by ion channels, which can be thought of electrically as conductors, as they facilitate electrical current flowing into or out of the cell. For a given ion i, the conductance of the ion by its associated ion channel can be expressed as some quantity g_i. We know that conductance is equal to current divided by voltage:

$$G = \frac{I}{V} \tag{5.4}$$

We can therefore model the current generated by a particular ion like this, for i the ion in question and V_m the membrane potential:

$$I_i = g_i(V_m - V_i) \tag{5.5}$$

Because sodium and potassium are the most important ions in the neuron that the squid giant axon is part of, Hodgkin and Huxley assumed that there would be two "voltage-gated channels" in the cell membrane (one for each of those two ions) that the neuron would use to actively regulate its ion concentrations, as well as a "leak channel" representing the ability of ions to passively diffuse across the cell membrane. (Note that other ions, such as calcium to give one example, are also important, and indeed other neuron models may use different ions than the Hodgkin-Huxley model does.) The leak channel was assumed to have constant conductance, while for the voltage-gated channels, the conductance was assumed to change based on how open the channel in question was. The calculations for the conductances of the sodium and potassium channels were approached differently. Hodgkin and Huxley assumed that the conductance for the potassium channel would be proportional to some variable n associated with the activation of that channel, and that the conductance of the sodium channel would be proportional to variables m and h associated with the activation and inactivation (respectively) of that channel. The three variables n, m, and h were all assumed to be on a scale from 0 to 1, and all of them are dimensionless quantities rather than concentrations of specific molecules that would activate or inhibit the ion channels. Hence, the assumption was that the neuron would regulate its ion concentrations by undergoing cellular processes that increase or decrease n, m or h. (These would correspond to terms in $\frac{dn}{dt}$, $\frac{dm}{dt}$, and $\frac{dh}{dt}$.) These processes are very complicated, so the actual differential equations in the model for $\frac{dn}{dt}$, $\frac{dm}{dt}$, and $\frac{dh}{dt}$ were assumed to vary directly based on the membrane

potential V_m rather than introducing more variables representing molecules involved in cell signaling. Specifically, functions α_n, β_n, α_m, β_m, α_h, and β_h (which all take as input V_m) were introduced to model the increases and decreases of n, m and h. Putting everything together yields the following model (note that $I = I_c + I_{\mathrm{K}} + I_{\mathrm{Na}} + I_l$ is the sum of the currents previously explained):

$$\begin{cases} I = C_m \frac{dV_m}{dt} + \bar{g}_{\mathrm{K}} n^4 (V_m - V_{\mathrm{K}}) + \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{\mathrm{Na}}) + \bar{g}_l (V_m - V_l) \\ \frac{dn}{dt} = (1-n)\alpha_n(V_m) - n\beta_n(V_m) \\ \frac{dm}{dt} = (1-m)\alpha_m(V_m) - m\beta_m(V_m) \\ \frac{dh}{dt} = (1-h)\alpha_h(V_m) - h\beta_h(V_m) \end{cases} \tag{5.6}$$

The functions causing the increases and decreases in the three dimensionless variables n, m and h were fitted by Hodgkin and Huxley to real data that they obtained in lab experiments. They were empirically determined to have the following forms, for V_r the resting potential of the neuron:

$$\begin{aligned} \alpha_n(V_m) &= \tfrac{0.01(10+V_r-V_m)}{\exp(0.1(10+V_r-V_m))-1} & \beta_n(V_m) &= 0.125\exp(\tfrac{V_r-V_m}{80}) \\ \alpha_m(V_m) &= \tfrac{0.1(25+V_r-V_m)}{\exp(0.1(25+V_r-V_m))-1} & \beta_m(V_m) &= 4\exp(\tfrac{V_r-V_m}{18}) \\ \alpha_h(V_m) &= 0.07\exp(\tfrac{V_r-V_m}{20}) & \beta_h(V_m) &= \tfrac{1}{\exp(0.1(30+V_r-V_m))+1} \end{aligned} \tag{5.7}$$

Note how the forms of the α and β functions are different for h compared to the other two, due to the slightly different biological role played by h. This system is 4-dimensional and highly nonlinear, so analytical solutions of it are impossible. However, a fixed point can be found, and the Jacobian of the system can be evaluated at it. Doing so reveals that there are two negative eigenvalues and two complex eigenvalues, which means that the preconditions for a Hopf bifurcation in the Hodgkin-Huxley model are satisfied. Most of the parameters in the model are biologically determined and therefore not easily manipulated, but taking the current I as a bifurcation parameter can cause the real parts of the two imaginary eigenvalues to switch from negative to positive. This Hopf bifurcation represents a change in model behaviour from the membrane potential reaching an equilibrium to the membrane potential exhibiting a periodic solution with sharp spikes in voltage. Biologically, these represent action potentials, also known as when the neuron is firing. The spikes in voltage bear almost no resemblance to sine waves. Periodic oscillations of this type, with sharp rather than gradual increases or decreases, are known as "pulse-relaxation oscillations" or simply "relaxation oscillations", following electrical engineering terminology. (If parameters are chosen such that the periodic solution does not exist, but the system is sufficiently close to the Hopf bifurcation, then the transient dynamics of V_m before reaching equilibrium will resemble one period of the periodic solution; biologically speaking, this is a single action potential.) Note that in general, pulse-relaxation oscillations occur when there is some separation in timescales between different variables in a model.

Owing to the success of the Hodgkin-Huxley model, many other mathematical neuron models have been proposed over the years. This includes models in which many neurons are coupled together in one system; since ions can be transmitted between neurons via

their synapses, the electrical potential in one neuron in such a system will depend on the electrical potential in any neurons that are upstream from it. For instance, a dynamical system modelling three neurons might look like this, for $i = 1, 2, 3$:

$$\begin{cases} I_i = C_m \frac{dV_{m,i}}{dt} + \bar{g}_{\mathrm{K}} n_i^4 (V_{m,i} - V_{\mathrm{K}}) + \bar{g}_{\mathrm{Na}} m_i^3 h_i (V_{m,i} - V_{\mathrm{Na}}) + \bar{g}_l (V_{m,i} - V_l) + I_{\mathrm{syn},i} \\ \frac{dn_i}{dt} = (1 - n_i)\alpha_n(V_{m,i}) - n_i \beta_n(V_{m,i}) \\ \frac{dm_i}{dt} = (1 - m_i)\alpha_m(V_{m,i}) - m_i \beta_m(V_{m,i}) \\ \frac{dh_i}{dt} = (1 - h_i)\alpha_h(V_{m,i}) - h_i \beta_h(V_{m,i}) \end{cases} \tag{5.8}$$

If we take $i = 1, 2, 3$, then this is a twelve-dimensional system, since each of the three neurons in the system has its own internal dynamics (and hence its own values for V_m, n, m, and h). However, the voltage in a given neuron may increase or decrease based on the voltages of other neurons that are connected to it via a synapse. Here, the term $I_{\mathrm{syn},i}$ for $i = 1, 2, 3$ (in other words, $I_{\mathrm{syn},1}$, $I_{\mathrm{syn},2}$, and $I_{\mathrm{syn},3}$) represents the amount of current that neuron i receives by virtue of having a synaptic connection with the other neurons. This is the way that electrical activity can propagate along many different neurons, causing signals to be transmitted through the entire nervous system.

Another important consequence of mathematical research on ODEs in neuroscience is the development of a way to reduce the differential equations for oscillatory quantities (i.e. any state variable in a dynamical system that has a periodic solution) to differential equations describing how far along in their periods these quantities are. (In other words, we would be describing the phase of each state variable.) This is well beyond the scope of this course, although the synchronization of voltages in different neurons as well as other oscillatory variables (i.e. determining when the differences in their phases go to zero) is an active area of current research.

5.2 December 1: SIR model; fitting a model to data

The event with the highest global impact in recent years is arguably the COVID-19 pandemic. Throughout the world, much of the response to Covid has been driven by mathematical models. Specifically, variations on one particular dynamical system model have been used to predict caseloads and deaths; this model is the SIR model. In addition to Covid, the SIR model has been used to make predictions regarding many other infectious diseases. Constructing this model from first principles is not very challenging. We start from the assumption that everyone in the population belongs to one of three categories, which are susceptible to infection, infected, and recovered from or resistant to infection (hence the name "SIR"). We therefore have three state variables S, I, and R, which represent the proportions of the population that belong to each group. The rates of change of each of these model components represent ways that a person can transition between categories, such as a susceptible person becoming infected or an infected person recovering. This style of dynamical system model, in which the state variables are categories of some kind, is called a "compartmental model".

So, how do we construct such a model? To start, we will look at one of the most basic cases: the standard SIR model without any added bells and whistles. We will assume that

the total population of wherever we're studying is constant, because population growth takes place over a longer timescale than disease spread. (In other words, we are assuming that total population is at equilibrium relative to S, I and R.) We will also assume only two events of interest in the system: a susceptible person catching the disease, and an infected person recovering. Assuming that the disease spreads by person-to-person contact, the rate at which a susceptible person can become infected is proportional to how often a person in the category S encounters someone in the category I. As we have previously seen with predator-prey models, this is an interaction term and hence scales with both S and I. Contact is also more frequent if the total population is smaller (and vice versa) as a smaller population means more opportunities for the same people to bump into each other. Taking the population size to be $S+I+R = N$, we get the interaction term describing the process of a susceptible person becoming infected to be $\frac{\beta}{N}SI$. This term is added to $\frac{dI}{dt}$ and subtracted from $\frac{dS}{dt}$, as it represents someone leaving the susceptible category and entering the infected category. Finally, we will assume that infected people recover at some rate γ. This means that the total number of people leaving the infected category and entering the recovered category per unit time is γI. Putting these together yields the following system:

$$\begin{cases} \frac{dS}{dt} = \frac{-\beta SI}{N} \\ \frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases} \tag{5.9}$$

One thing that we can immediately notice (and that we assumed during the derivation of the model) is that the total population is some constant N, and hence a conserved quantity. Therefore, we can reduce the dimensionality of this system by 1, which we can easily do by considering $R = N - S - I$ since R does not occur in any of the three ODEs making up the model. Taking $N = 1$, as is often done, means that the state variables represent percentages of a total population. This also means that the model does not output fractional numbers of people, although the presence of this behaviour is not typically viewed as a problem since all models are approximations in the first place. We therefore can get the following, the simplest form of the SIR model:

$$\begin{cases} \frac{dS}{dt} = -\beta SI \\ \frac{dI}{dt} = \beta SI - \gamma I \end{cases} \tag{5.10}$$

This is simpler than the Hodgkin-Huxley neuron model, or even the Lotka-Volterra predator-prey model, and we can determine a lot about its behaviour analytically. For instance, we can find its fixed points. Based on the equation for $\frac{dI}{dt}$, we need $\beta SI = \gamma I$, which is true if $I = 0$ or $S = \frac{\gamma}{\beta}$. However, if $I \neq 0$, then we get a nonzero value for $\frac{dS}{dt}$ (and also $\frac{dR}{dt}$). This means that $I = 0$ is a requirement for a fixed point of this model, but there are no other requirements, so $(S^*, I^*, R^*) = (k, 0, 1-k)$ is a fixed point for any $k \in [0, 1]$. The interpretation of this is that the epidemic will stop when there are no more infectious people to spread the disease, which is very intuitive.

One important part about the model is the rate of change of the infected component, since that will determine whether the epidemic under consideration will spread or die out. To examine this, we will consider a population that starts out entirely susceptible (i.e.

$S(t=0)=1$). Now, suppose that a disease is stochastically introduced into this population. In other words, suppose that we perturb S and I by $-\varepsilon$ and $+\varepsilon$, respectively, where ε is a number very close to zero. This means that S is still approximately 1, but I is now nonzero and the rates of change in the model involving I are also nonzero. In particular, we now have the following for the rate of change of I:

$$\frac{dI}{dt} \approx (\beta - \gamma)I \tag{5.11}$$

From this, we can tell that the infection will spread if $\beta > \gamma$, since $\frac{dI}{dt}$ will be positive in that case. Likewise, if $\gamma > \beta$, the infection will die out, as the rate at which people get infected will be less than the rate at which infected people recover. Since 2020, you will have undoubtedly heard the term "R-naught", or "R_0", many times in describing how capable a disease is of spreading. It is actually a term taken directly from this model:

$$R_0 = \frac{\beta}{\gamma} \tag{5.12}$$

As you may already know, a value of R_0 above 1 means that a disease will spread, while a value of R_0 below 1 means that it will die out. This makes R_0 the most important quantity represented in the model; in epidemiology, determining R_0 for a given disease is often the main goal of research. Another important feature of the model is that since γ represents the rate at which infected individuals recover per unit time, the quantity γ^{-1} represents the average length of time that a person stays infected for. Likewise, βS can be thought of as the number of people that an infected person can themselves infect per unit time, and hence an interpretation for β is the number of contacts an infected person has over a given length of time multiplied by the probability of each contact becoming infected. This means that both β and γ can be input into the model from real data, which I'll explain more about later.

One important thing about the SIR model is the fact that its simplicity makes it endlessly customizable. Diseases are very heterogeneous in terms of their effects and characteristics, and this can be reflected by adding new terms to the basic SIR model. For example, suppose that recovery from a particular disease does not grant immunity to that disease going forward. This can be reflected by turning the SIR model into what might be referred to as an SIS model:

$$\begin{cases} \frac{dS}{dt} = -\beta SI + \gamma I \\ \frac{dI}{dt} = \beta SI - \gamma I \end{cases} \tag{5.13}$$

This is even simpler, and can actually be solved analytically after reducing the model to one dimension based on the conserved quantity $S + I = 1$. (You can do this on your own time if you want.)

So far, we have assumed that nobody actually dies from the disease, which is obviously an inaccurate assumption to make. What if they do? This can be represented by adding a new state variable D (for "dead"), and an additional term to $\frac{dI}{dt}$:

$$\begin{cases} \frac{dS}{dt} = -\beta\frac{SI}{N} \\ \frac{dI}{dt} = \beta\frac{SI}{N} - \gamma I - \mu I \\ \frac{dR}{dt} = \gamma I \\ \frac{dD}{dt} = \mu I \end{cases} \tag{5.14}$$

Here, μ is (as you might be able to guess) the mortality rate of the disease. Note that this version of the model also explicitly brings back $N = S + I + R$, since the assumption that the population is constant over time is now broken.

Alternatively, suppose that when someone is exposed to a particular disease, there is some latency period during which they do not show symptoms of the disease and cannot infect other people. This additional state can be accounted for by introducing another state variable (E for "exposed") to the model. We could then end up with something like this:

$$\begin{cases} \frac{dS}{dt} = -\beta\frac{SI}{N} \\ \frac{dE}{dt} = \beta\frac{SI}{N} - \alpha E - \gamma E \\ \frac{dI}{dt} = \alpha E - \mu I - \gamma I \\ \frac{dR}{dt} = \gamma E + \gamma I \\ \frac{dD}{dt} = \mu I \end{cases} \tag{5.15}$$

Note that here, the calculations that go into finding R_0 will be quite different compared to what they are in the simple SIR model. Additionally, further possibilities for building a model are essentially endless. (That's one of the big issues in mathematical modelling: knowing which of the millions of possible model formulations to use.)

So, how do we use such a model in practice? We have all of these parameters that we need in order to determine the exact dynamics of the model, so how do we find their values? The answer is based on the available data. If we are trying to determine how an actual disease will spread throughout an actual population, we might have data points representing case counts, deaths and recoveries over time in that population. This gives us a set of points that our model should replicate at least fairly accurately, if it is a good model for predicting future dynamics of the same disease in the same population. We can therefore pick several sets of values for the model parameters, then simulate the chosen parameter sets in the model, to see which set corresponds to the best fit of the model output for the real-life data. (This can be done using least squares, for example.)

But how do we choose the parameter values to test? One way to do this is by random sampling. If our model doesn't have very many parameters that we need to determine, then we can simulate all combinations of values within whatever ranges we think those parameters are likely to be. For instance, the original SIR model has only two parameters, β and γ. If (for example) we thought that β was between 2 and 3, and γ was between 0.5 and 1.5, then we could test value pairs (β, γ) featuring regularly spaced values of β in the interval $[2, 3]$ (e.g. 2, 2.1, 2.2, et cetera), and likewise for $\gamma \in [0.5, 1.5]$. However, if we have a lot of parameters to fit, then this will take a lot of time. One alternative is to use what is called "Latin hypercube sampling", which significantly cuts down on computation time, and is easy to implement (it's actually a bit like sudoku). Suppose that for each parameter in our model, we have n possible choices of values for that parameter, which we

can number from 1 to n. (So, in the SEIRD model above with parameters β, α, γ and μ, we would have $\beta_1, \ldots, \beta_n$ being the potential choices for β, $\alpha_1, \ldots, \alpha_n$ being the choices for α, and likewise for γ and μ.) A Latin hypercube sample is one in which we take n different parameter sets overall, so that each value from 1 to n only occurs once for each parameter. So, for instance, if we had a standard SIR model with two parameters β and γ, and four different choices for each parameter, then one possible Latin hypercube sample would be $\{(\beta_1, \gamma_2), (\beta_2, \gamma_4), (\beta_3, \gamma_3), (\beta_4, \gamma_1)\}$. Another advantage of a Latin hypercube sample is that it portrays a relatively accurate picture of the variability in parameters; sampling four points (β, γ) at random might lead to something like $\{(\beta_3, \gamma_3), (\beta_3, \gamma_4), (\beta_4, \gamma_3), (\beta_4, \gamma_4)\}$ where the samples are all clustered together.

But how do we know these ranges that we can choose from? The best way to do this is do draw values directly from the literature related to the problem that we're trying to model. This is where knowledge of some subject besides math comes in handy, since building a model will invariably involve reading papers published in different fields. However, this is well beyond the scope of this course.

5.3 December 3: Blood glucose model

Another field that mathematical models are often used in (that we haven't seen very many examples from yet) is physiology. The human body can be thought of as a vast network of interacting processes, in which different types of cells and chemicals are produced, perform their functions, and are consumed or destroyed. You've seen with the SIR model that dynamical system models can be constructed based on diagrams of boxes and arrows, in which the boxes represent different categories of people (susceptible to a disease, currently infected, recovered from the disease, et cetera) and the arrows represent the movement of people between categories. In physiology, similar principles are used in order to construct mathematical models from process diagrams. The simplest form of this, which you've already seen, is reaction kinetics, in which an enzyme converts its substrate into a reaction product. However, many biological processes are not as simple as one chemical being turned into another. For instance, in the human pancreas, there exists a certain kind of cell called the β cell whose job it is to produce insulin, but this is obviously not the same as actual β cells being converted into insulin. In general, this means that we can still start off a mathematical model in physiology by drawing boxes and arrows, but the boxes and arrows in question might have different meanings than we previously used in the SIR model.

Since the human body is so complex, most mathematical models used in physiology cover specific processes. Models on a larger scale do exist: for instance, Prof. Robert Hester at the University of Mississippi works on a very large mathematical model called HumMod, which simulates a variety of different physiological processes on the scale of the entire human body. The full range of processes in that model obviously cannot be fully explained within the span of one lecture, so I will instead focus on something on a smaller scale, namely a model of insulin production that has been used to predict the onset of diabetes. Since diabetes is associated with elevated blood glucose levels and decreased insulin production, the model tracks three different variables: blood concentrations of glucose and insulin, and mass of β cells (which produce insulin). This means that the model will look like this, for G glucose

concentration, I insulin concentration, and β the mass of β cells in the pancreas:

$$\begin{cases} \frac{dG}{dt} = u_1(G, I, \beta) \\ \frac{dI}{dt} = u_2(G, I, \beta) \\ \frac{d\beta}{dt} = u_3(G, I, \beta) \end{cases} \tag{5.16}$$

In order to get the specific terms in the model, we can think all the way back to the tank problems that we saw at the beginning of the course, where the rate of change of the contents of a tank was defined as the rate in minus the rate out. Likewise, the rate of change for blood glucose will be the rate at which it is produced minus the rate at which various parts of the body uptake it for their own use:

$$\frac{dG}{dt} = \text{Production} - \text{Uptake} \tag{5.17}$$

Since we are dealing with an actual physical quantity (i.e. glucose concentration), it is highly important to get the units right for G, $\frac{dG}{dt}$, and the terms defining $\frac{dG}{dt}$. (If we built all of the terms in the model correctly, but we evaluated the model dynamics where G is on some highly implausible scale such as millions of kilograms per liter, then we wouldn't get any results that are biologically meaningful.) A common scale used to track glucose concentration in medical settings is milligrams per deciliter, so we will take those to be the units for G. Additionally, this model specifically looks at changes in fasting glucose levels over long time scales (days to years), we will measure time t in days. Therefore, the units for $\frac{dG}{dt}$ are milligrams per deciliter per day, and therefore all terms that make up the differential equation $\frac{dG}{dt}$ must be in terms of milligrams per deciliter per day as well.

So, what factors influence the production and uptake of glucose? Based on experimental data, we know that these rates are based on the concentrations of insulin and glucose itself in the blood. When insulin concentration is held constant, increasing blood glucose causes less additional glucose to be produced by the body, and more of the glucose in the blood to be uptaken. This means that $\frac{dG}{dt}$ might take the form $a - \sigma G$, for a the net rate of glucose production when blood glucose levels are zero, since higher levels of G put downward pressure on G. (This is known as a "negative feedback loop"; a "positive feedback loop" is when higher levels of one variable cause the rate of change of that variable to increase even further.) We also know that higher blood insulin concentration causes greater uptake of glucose; diabetes is associated with the breakdown of this relationship, where the body's sensitivity to insulin is low. We can thus formulate $\frac{dG}{dt}$ as follows:

$$\frac{dG}{dt} = a - (b + cI)G \tag{5.18}$$

Here, b is the rate at which glucose is utilized by the body independent of insulin concentration, and c is insulin sensitivity. By our assumptions, the units for a need to be the same as those for $\frac{dG}{dt}$, namely milligrams per deciliter per day. Since the term $(b + cI)$ multiplies G, it needs to be in terms of day^{-1} so that the units line up. Therefore, b has units of day^{-1}, and the units of c will be whatever is needed to cancel out the units of I (see below) to get cI also on the scale of day^{-1}.

For the differential equation governing insulin, $\frac{dI}{dt}$, we can start from similar assumptions. First of all, insulin concentration is often measured in thousandths of international units of

insulin per milliliter, or μU ml^{-1}. Since we already know that we will be measuring time in days, that gives us units of μU ml^{-1} day^{-1}. We know that insulin is secreted by β cells, and that it is cleared by being uptaken by the liver, kidneys and various insulin receptors. We can therefore assume the following "rate in minus rate out"-style dynamics for $\frac{dI}{dt}$:

$$\frac{dI}{dt} = \text{Secretion} - \text{Clearance} \tag{5.19}$$

We know that β cells secrete insulin in response to high levels of blood glucose concentration. More specifically, experimental work has shown that the relationship between glucose concentration and the rate at which insulin is secreted by β cells is sigmoidal. We will therefore assume that the secretion term in $\frac{dI}{dt}$ is a sigmoidal saturation function of G (which will also depend on how many β cells there are). One relatively standard form for a sigmoidal function is $u(x) = \frac{x^2}{1+x^2}$. We will additionally assume that the maximum rate of insulin production is some constant d, and that the half-saturation constant that affects the shape of the sigmoidal curve is some value e. Furthermore, we will also assume that insulin is cleared at a constant rate f. This yields the following form for $\frac{dI}{dt}$:

$$\frac{dI}{dt} = \frac{d\beta G^2}{e + G^2} - fI \tag{5.20}$$

Once again, we will need our parameters to take units so that all terms in $\frac{dI}{dt}$ are measured in μU ml^{-1} day^{-1}. It should be relatively obvious that f has the units of day^{-1}. You can figure out the units for the rest of the parameters yourself (note that β measures the mass of existing β cells and is measured in milligrams).

What about $\frac{d\beta}{dt}$? Since β represents the mass of β cells, we can assume that the rates of change of β are related to β cells replicating and dying. We know that the replication of β cells increases with blood glucose concentration, for the same reasons that blood glucose concentration increases the rate of insulin production by existing β cells. However, extremely high levels of blood glucose concentration have been shown experimentally to decrease β cell replication. We will therefore assume that the rate at which new β cells are produced is a quadratic function of G, something like $\alpha G - \gamma G^2$, making the mass of β cells produced per unit time something like $(\alpha G - \gamma G^2)\beta$. As for β cell death, this can happen in two different ways, namely apoptosis (planned, or "natural", cell death) and necrosis (unregulated cell death due to harmful conditions). Experimental results suggest that β cell death also varies nonlinearly with glucose concentration, albeit in the opposite directions compared to replication. Therefore, we will assume that the death rate for β cells is another quadratic polynomial in G, namely $k - \delta G + \eta G^2$. This makes the mass of β cells that die over a given time interval equal to $(k - \delta G + \eta G^2)\beta$, and hence the β cell death term will be $-(k - \delta G + \eta G^2)\beta = (-k + \delta G - \eta G^2)\beta$. Adding these together gives us our equation for $\frac{d\beta}{dt}$:

$$\frac{d\beta}{dt} = (-k + hG - mG^2)\beta \tag{5.21}$$

Therefore, we get the following for our model:

$$\begin{cases} \frac{dG}{dt} = a - (b + cI)G \\ \frac{dI}{dt} = \frac{d\beta G^2}{e+G^2} - fI \\ \frac{d\beta}{dt} = (-k + hG - mG^2)\beta \end{cases} \tag{5.22}$$

So, what can we do with this model? For one, we can determine which parameter values are likely to lead to normal versus diabetic blood glucose levels. This can be demonstrated by finding fixed points in the model, which we can do analytically. There exists one which corresponds to healthy levels of G, I and β, one that represents a hyperglycemic state in which G is pathologically high and I and β are both zero, and one additional fixed point between them. The pathological equilibrium is just $(G^*, I^*, \beta^*) = (\frac{a}{b}, 0, 0)$, which can be obtained trivially. The other two take the form $(G^*, I^*, \beta^*) = (G_i, I_i, \beta_i)$, for $i = 1, 2$ and the following specific parameter values:

$$G_{1,2} = \frac{h \pm \sqrt{h^2 - 4mk}}{2m} \tag{5.23}$$

$$I_i = \frac{a}{cG_i} - \frac{b}{c} \tag{5.24}$$

$$\beta_i = \frac{fI_i(e + G_i^2)}{dG_i^2} \tag{5.25}$$

The Jacobian can be calculated for each of these points, and the stability of them can therefore be determined. When taking parameter values corresponding to experimental results, the healthy and pathological fixed points are both stable, while the fixed point between them is a saddle point. However, if we perturb different model parameters, we can cause bifurcations to happen. For instance, if we decrease h, corresponding to a drop in β cell production, the healthy fixed point and the saddle point move closer together. For a critical value of h, these two fixed points collide and cause a fold bifurcation, annihilating one another. The effect of this is that if h falls below that critical threshold, the only equilibrium will be the pathological equilibrium. Another important analysis of this model has been to incorporate periodic oscillations in blood glucose and β cell count. The original version of the model always considered fasting levels of glucose, insulin and β cells, whereas in real life, these levels may vary over the course of the day. Allowing different model parameters to change based on the human circadian rhythm caused much more complicated dynamics to emerge, including additional ways for hyperglycemic conditions to develop. The research on this was actually just recently performed by a friend of mine, and the paper containing the important findings is still under review, so I can't really reveal too much more about it. However, there are still many more results pertaining to diabetes waiting to be discovered, which almost certainly include some that can be obtained by using mathematical models.

5.4 December 5: Spatially explicit systems of ordinary differential equations

Since Calculus 3 is a corequisite for this course, and this is the last lecture that I'm giving this semester, I'm sure that you know all about partial derivatives by now. Therefore, you might think that we can construct differential equations using partial derivatives as well. This is correct; these are known as partial differential equations. For instance, you might run into this one a lot:

$$\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2} \tag{5.26}$$

Or this one:

$$\frac{\partial^2 u}{\partial t^2} = k\frac{\partial^2 u}{\partial x^2} \tag{5.27}$$

As this is a class on ordinary differential equations rather than partial ones, the analysis of these is outside the scope of it. However, it is certainly possible to represent spatial patterns within a framework of ordinary differential equations. Instead of representing space as another variable, this is typically done by assuming different copies of the same dynamical system, representing the dynamics in several different locations. The state variables in these systems then influence each other based on how the objects that they represent interact across space. I will show you some examples of this, based on models that you have already seen.

Let's start with a simple one. Suppose we have a predator-prey system, specifically the Lotka-Volterra model that we saw in class earlier:

$$\begin{cases} \frac{dN}{dt} = rN - \alpha NP \\ \frac{dP}{dt} = \beta NP - mP \end{cases} \tag{5.28}$$

What if we have two different locations that the predators and prey both live in? We can first extend this model to represent the predators and prey in both locations:

$$\begin{cases} \frac{dN_1}{dt} = r_1 N_1 - \alpha_1 N_1 P_1 \\ \frac{dP_1}{dt} = \beta_1 N_1 P_1 - m_1 P_1 \\ \frac{dN_2}{dt} = r_2 N_2 - \alpha_2 N_2 P_2 \\ \frac{dP_2}{dt} = \beta_2 N_2 P_2 - m_2 P_2 \end{cases} \tag{5.29}$$

Note that I have introduced the indices 1 and 2 for the predators and prey in locations 1 and 2. I also use these indices for the parameters which represent conditions in each location that might differ. For example, if we think of the prey as a herbivore, which eats grass, the local prey growth rates r_1 and r_2 might be different depending on which area is more suitable for grass to grow.

So far, with this setup, the predators and prey in each location (also called a "patch") are completely independent of each other. However, in real life, one or both species might migrate between patches. Suppose that over some unit of time, the proportion of prey that

migrate from one patch to the other is μ_N, which can be taken to be some value between 0 (no migration happens at all) and 1 (the entire population migrates during that length of time). It could theoretically also be greater than 1 if the time it takes for the entire population to move between patches is less than what t is defined as. Likewise, suppose that some proportion of predators also moves between patches over the unit of time. We will call this μ_P. This allows us to rewrite our system to include the effects of migration:

$$\begin{cases} \frac{dN_1}{dt} = r_1 N_1 - \alpha_1 N_1 P_1 + \mu_N N_2 - \mu_N N_1 \\ \frac{dP_1}{dt} = \beta_1 N_1 P_1 - m_1 P_1 + \mu_P P_2 - \mu_P P_1 \\ \frac{dN_2}{dt} = r_2 N_2 - \alpha_2 N_2 P_2 + \mu_N N_1 - \mu_N N_2 \\ \frac{dP_2}{dt} = \beta_2 N_2 P_2 - m_2 P_2 + \mu_P P_1 - \mu_P P_2 \end{cases} \tag{5.30}$$

The functional form for migration that I have used here is often called "passive dispersal", because it is essentially identical to diffusion or Brownian motion. We have also made the assumption here that the rate of migration from Patch 1 into Patch 2 is the same as the rate of migration from Patch 2 into Patch 1 for both species. This is not necessarily true: if a species prefers one patch or the other, the migration rates will not be symmetric. We could then include terms like $\mu_{N_{1,2}}$, $\mu_{N_{2,1}}$, $\mu_{P_{1,2}}$, and $\mu_{P_{2,1}}$ to describe the specific rates of migration from one given patch to another.

So, what are the effects of migration in this system? Well, consider the case without it, but where the parameters in each patch are different. (So, assume that $r_1 \neq r_2$, et cetera.) In such a system, the dynamics of the predators and prey in the two different locations will be completely different, with the two pairs of solutions (N_1, P_1) and (N_2, P_2) each following their own trajectories. However, increasing the migration parameters (each μ) will cause the solutions in each patch to resemble each other more and more. The reason for this is that we have added a term $\mu_N(N_2 - N_1)$ to N_1 that subtracts a proportion of the prey in patch 1 and adds a proportion of the prey in patch 2 (the other patch), and so on for P_1, N_2 and P_2. As these proportions increase towards 0.5, the effect becomes more and more like subtracting half of N_1 from N_1 and adding back half of N_2, which has a similar effect as averaging the populations in the two patches.

We also know that populations in the Lotka-Volterra model tend to oscillate, due to the limit cycle that exists in the (N, P)-plane. What happens when we introduce migration? If the values of the μ parameters are high enough, the oscillations of N_1 and N_2 will start to synchronize, and likewise with P_1 and P_2. This happens even if their frequencies had been very different in the absence of migration due to different parameter values in patch 1 and patch 2. In fact, the more the solution trajectories in each patch differ from each other in terms of shape, the higher the migration threshold above which synchrony occurs.

What if we have even more patches? (Real-life ecological networks are often fairly large.) Then, we could assume that each species in our system can migrate between some or all of the pairs of patches, and derive equations like this:

$$\begin{cases} \frac{dN_i}{dt} = r_i N_i - \alpha_i N_i P_i + \sum_{j \neq i} \mu_{N_{i,j}} N_j - \mu_{N_{j,i}} N_i \\ \frac{dP_i}{dt} = \beta_i N_i P_i - m_i P_i + \sum_{j \neq i} \mu_{P_{i,j}} P_j - \mu_{P_{j,i}} P_i \end{cases} \tag{5.31}$$

Note that some of the μ constants might be zero, if there is no migration between one or more pairs of patches. For instance, two patches might be separated by impassable terrain such as a mountain range or a large body of water. Additionally, we might consider a case in which migration only happens one way, such as populations in a river. Since the flow of the river only goes one way, we might have something like $\mu_{N_{1,2}} = 0$ but $\mu_{N_{2,1}} \neq 0$. In other words, patch 1 can send organisms to patch 2, but cannot receive organisms from patch 2. In general, this framework allows us to construct arbitrarily large networks with arbitrarily many patches, of which each might have its own different parameter values corresponding to different ecological conditions. As a result, we can model spatially complex scenarios without leaving the boundaries of ODEs.

For another example, in a previous lecture, I alluded to the fact that the Hodgkin-Huxley neuron model can be used to simulate the dynamics in multiple different neurons that are connected by synapses. In that version of the Hodgkin-Huxley model, we assumed that the variables in each neuron that represented activation or inactivation of the neuron's ion channels would not depend on the conditions in other neurons. (In other words, their dynamics would be local.) On the other hand, we assumed that the voltage in a given neuron would be affected by the voltages in the neurons that it is connected to. This is because there is a biological mechanism that allows voltage to be carried from one neuron to the next, namely the release of neurotransmitters, while there is no biological mechanism that would cause the ion channels in one neuron to open and close based on whether the ion channels in a different neuron are open or closed.

As it turns out, we can actually model the effects of specific synapses and neurotransmitters. Synapses can either be excitatory or inhibitory: an excitatory synapse increases the voltage in the postsynaptic neuron, increasing the possibility of a spike, whereas an inhibitory synapse decreases voltage in the postsynaptic neuron and reduces the possibility of it spiking. Regardless of whether the synapse is excitatory or inhibitory, the current produced by it follows the same general pattern, similar to the current produced by an ion channel:

$$I_{\text{syn}} = \bar{g}_{\text{syn}} f(V_m, \ldots)(V_m - V_{\text{syn}}) \tag{5.32}$$

In other words, the current produced by a given synapse has a maximum value $\bar{g}_{\text{syn}}$, and depends on the reversal potential V_{syn} of the neurotransmitter that the synapse uses. Apart from this, synapses can get very biologically complex (hence the function f that I have left unknown). Explaining these would take a long time to do, and the biology involved is probably beyond the scope of this course, so I'll leave the problem of constructing a model of multiple neurons as an exercise for those interested.