Mathematical Biology - Week 1

October 5, 2022

1 Introduction

Mathematical biology is exactly as it sounds – it entails the use of mathematics to describe biological phenomena in order to better understand biological systems, as well as predict their behaviour. It is an incredibly diverse field aimed at utilising the complete mathematical toolbox to ascertain insight into many areas of biology.

The aim of this course is to provide a foundation in mathematical biology through the mathematical models commonly used to describe biological processes. The course will rely primarily on tools from applied dynamical systems, applied PDEs, asymptotic analysis, and stochastic processes. The course will be broadly organised around mathematical topics. After a discussion of the background mathematical theory for each topic, the models of biological processes that fit under the topic areas will the introduced and analysed.

We will begin with the simplest models that are one-dimensional systems, or first-order ODEs, which can describe biochemical reaction dynamics, population dynamics and epidemiological phenomena. We will see that these models are the building blocks from which the more complex models are constructed. The increasing complexity introduces the need for additional mathematical tools. For example, models of chemical reaction networks are obtained by introducing interaction terms that couple two or more first-order ODEs. This results in a multi-dimensional system, or system of ODEs. The incorporation of spatial dynamics into population models changes the original ODE to a PDE. Finally, the introduction of variability requires formulating a stochastic process whose average dynamics will sometimes coincide with those given by the ODE model. If this is all a bit unclear at the moment, don't worry – we'll be going over all of the details and, hopefully, in retrospect, you'll see how the topics of the module are connected.

1.1 Organisation and delivery

The content of the module is divided into 5 parts

1. One-dimensional systems
2. Multi-dimensional systems I: Fixed points and Stability
3. Multi-dimensional systems II: Bifurcations and Oscillations
4. Spatial dynamics
5. Stochastic processes

and each section on average should take around 2 weeks to complete. There will be one set of problems for each section. The lecture notes and problem sets will be distributed in the form of Jupyter notebooks, though PDF versions will also be provided. The use of Jupyter notebooks is to allow you to generate plots and visualisations of the solutions to the models by running short segments of code that are embedded within the notebooks. This will complement the analytical approaches that we will discuss and lead to a better understanding and interpretation of the results.

1.1.1 Problems Classes and office hours

Approximately every two weeks there will be a problems class that will begin with some basic content from the problem sheets and lectures with the aim of expanding to a broader discussion about the material. Office hours will be held in person in my office, 741 Huxley.

1.2 Books

The lecture notes were assembled using several sources. These include the 'core' books listed on Legato: 1. *Mathematical Biology I: An Introduction* by JD Murray 2. *Mathematical Biology II: Spatial Models and Biomedical Applications* by JD Murray 3. *Nonlinear Dynamics and Chaos* by S Strogatz 4. *An Introduction to Stochastic Processes with Applications to Biology* by LJS Allen

along with the additional texts

1. *Mathematical Models in Population Biology and Epidemiology* by F Brauer and C Castillo-Chavez
2. *Mathematical Modelling of Biological Processes* by A Friedman and C-Y Kao

While the notes are intended to be comprehensive, I encourage you to crack open these books and read what they have to say about the material that you are learning. The texts could provide you with different perspective on the material and give you a different angle from which to understand it. This is always useful!

2 One-dimensional or first-order systems

In the first part of this module, we will discuss biological systems that can be modelled as a one-dimensional or first-order system which is an ODE of the form

$$\begin{aligned} \frac{dx}{dt} &= f(x), \\ x(0) &= x_0, \end{aligned} \tag{1}$$

where $x(t)$ is a real-valued function of time, t, and $f(x)$ is a sufficiently smooth real-valued function of x. These notes largely summarise Chapter 2 of Strogatz.

Before discussing and describing models that can be described by (1), it is important to ensure that we are equipped with the tools that we need to analyse these systems. My hope is that you encountered them already in other modules, so this will largely be a review. I will also assume that you are familiar with solving linear differential equations. Hopefully, one-dimensional systems will be a familiar mathematical environment to introduce some of the common features and building blocks of the models used to describe biological processes. We will see that these features appear again and again, though in more mathematically-complex scenarios.

2.1 Existence and Uniqueness

In constructing a model, it is often useful to know whether it is well-posed and you can be guaranteed the existence of a unique solution. As it turns out, it can be provided $f(x)$ is sufficiently smooth.

Existence and Uniqueness Theorem: Consider the initial value problem given by (1). Suppose that $f(x)$ and $f'(x)$ are continuous on an open interval R of the x-axis, and suppose that x_0 is a point in R. Then the initial value problem has a solution $x(t)$ on some time interval $(-\tau, \tau)$ about $t = 0$, and the solution is unique.

The one-dimensional system

$$\begin{aligned}\frac{dx}{dt} &= x^{1/3},\\ x(0) &= 0,\end{aligned}$$

is not unique as $f'(x)$ is not defined at $x = 0$. It can be shown that both $x = 0$ and $x = (2t/3)^{3/2}$ are solutions.

An important aspect of this theorem is the fact that it only applies over the interval $(-\tau, \tau)$ and not necessarily for all time. For example, the initial value problem

$$\begin{aligned}\frac{dx}{dt} &= 1 + x^2,\\ x(0) &= 0,\end{aligned}$$

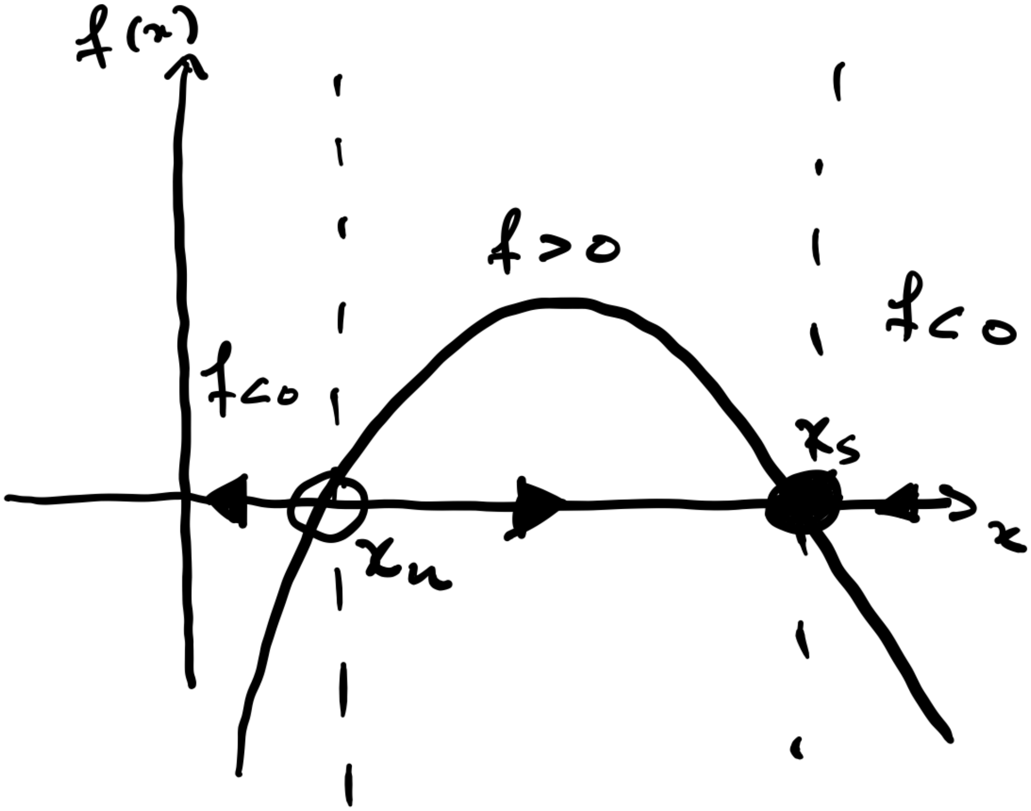
has solution $x(t) = \tan t$, and thus the solution only exists for $-\pi/2 < t < \pi/2$.

2.2 Fixed points and their stability

As will often be the case for nonlinear systems, we will not be able to find analytical solutions to the equations and we will have to resort to other means to analyse their behaviour. One approach is to identify particular values of x and ascertain whether the solution will move toward these values as $t \to \infty$, or whether it will move away from them, even if the initial value, x_0, is arbitrarily close.

2.2.1 Fixed points

These special points that we will identify are called **fixed points**, and are the values of x for which $dx/dt = 0$, or, given (1), the values of x for which $f(x) = 0$. Fixed points can also be referred to as steady states. Thus, if the initial condition, x_0, is a fixed point, the solution is $x(t) = x_0$. This seems like the most boring points to examine, but their importance in thinking about the system is perhaps best illustrate graphically.



Before discussing fixed points, it is useful to introduce some terminology. On the x-axis, we have drawn arrows to indicate the direction in which a point will move based on the sign of $f(x)$ at that point. In this scenario, we refer to these points as **phase points**, their motion as a **flow**, their path as **trajectories**, and finally the x-axis and the arrows as the **phase portrait**.

Now, back to fixed points. We see in the illustration there are two fixed points x_U and x_S. We see that $f(x) < 0$ for $x < x_U$ and $f(x) > 0$ for $x_U < x < x_S$. Therefore according to (1), if the initial condition satisfies $x_0 < x_U$ it will move to the left (as $f(x) < 0$), while if $x_U < x_0 < x_S$ it will move to the right (as $f(x) > 0$). In both cases, the solution will move away from x_U. As a result, x_U is referred to as an **unstable** fixed point. As you might have guessed, we have the opposite situation for x_S where initial conditions nearby will stay in the vicinity of x_S or move towards x_S. In this case, we have a **stable** fixed point. If the trajectories approach a stable fixed point as $t \to \infty$, we call this fixed point **asymptotically stable**. We can also have **half-stable** fixed points as in the case of $x = 0$ for $f(x) = x^2$.

Before continuing with the discussion of stability, it's worth noticing that the illustration suggests monotonic behaviour. Namely, if the solution moves to the right at any point in time, it will continue to do so. This inability to oscillate is a consequence of the uniqueness of the solution. If it were to back track along the line, it would automatically mean that $f(x)$ is multivalued, thereby violating our requirements on continuity.

2.2.2 Linear stability analysis

We have just seen graphically how to identify the stability of fixed points, but there is also a more formal approach called **linear stability analysis** and it is carried out as follows:

Suppose the point x^* is a fixed point. To evaluate its linear stability, we linearise $f(x)$ about x^* via a Taylor expansion

$$f(x) = f(x^*) + (x - x^*)f'(x^*) + \text{h.o.t.}$$

We first recognise that $f(x^*) = 0$ by virtue of the fact that x^* is a fixed point. Defining $\eta = x - x^*$ as the perturbation from x^*, the differential equation $\mathrm{d}x/\mathrm{d}t = f(x)$ linearised about the fixed point x^* is then

$$\frac{d\eta}{dt} = f'(x^*)\eta. \tag{2}$$

Thus, we see that:

1. $f'(x^*) > 0$, then the perturbation η will grow exponentially with time. This means that x^* is an unstable fixed point.
2. $f'(x^*) < 0$, then the perturbation η will decay exponentially with time. In this case, x^* is a stable fixed point.

For the case where $f'(x^*) = 0$, the stability cannot be determined without examining the higher-order terms which are themselves nonlinear. For example, consider $f(x) = x^3 + c$. Linear stability can be used if $c = 1$, but not if $c = 0$. Thus, linear stability analysis is not all encompassing but usually a useful tool nonetheless.

2.3 Bifurcations

In the previous sections, we discussed how to identify fixed points, x^*, by solving the equation $f(x) = 0$ and assessing their linear stability by evaluating $f'(x^*)$. Remember, the fixed points are important as they tell us where the solution wants to move towards or away from.

What we'd like to examine now is how these fixed points and their stabilities change as we vary parameters that appear in the function $f(x)$. In the context of mathematical biology, these parameters might be the rate at which a chemical reaction occurs, or perhaps the value at which a population will saturate. We'll look at examples of both in the near future. When changes in the parameter values lead to qualitative changes in the fixed points and their stability, e.g. a change in the parameter resulted in the number of fixed points going from 1 to 3, we say a **bifurcation** has occurred. The parameter values at which we see the bifurcation is called the **bifurcation point**. In one-dimensional systems, there are three types of bifurcation that can occur, each of which is described using their *normal forms*, the form which describes the bifurcations of a given type near the bifurcation point.

2.3.1 Saddle-node bifurcation

With a saddle-node bifurcation, a pair of fixed points (one stable and the other unstable) are either created or destroyed as the control parameter is varied. The normal forms for this type of bifurcation are

$$\frac{dx}{dt} = r + x^2, \tag{3}$$

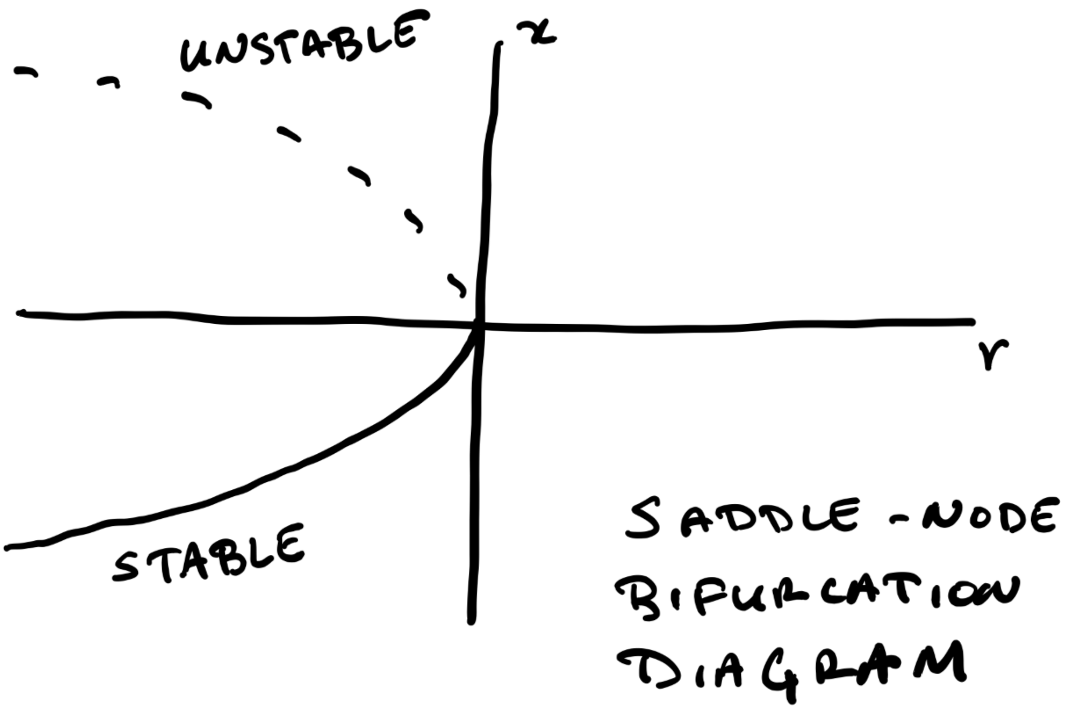
and

$$\frac{dx}{dt} = r - x^2, \tag{4}$$

where r is the control parameter. As you can see, varying r can raise or lower the parabola and hence control whether or not it crosses the x-axis.

In (3), we only obtain real solutions to $x^2 = -r$ when $r \leq 0$. Thus there are no fixed points when $r > 0$. The bifurcation occurs at the bifurcation point $r = 0$ when there is a single fixed point that is half-stable. For $r < 0$, there are 2 fixed points at $x^* = \pm\sqrt{|r|}$ with $-\sqrt{|r|}$ being stable and $\sqrt{r}$ being unstable.

The bifurcation and resulting fixed points for different values of r and x can be compiled in the form of a bifurcation diagram. For the saddle-node bifurcation described here, the bifurcation diagram is



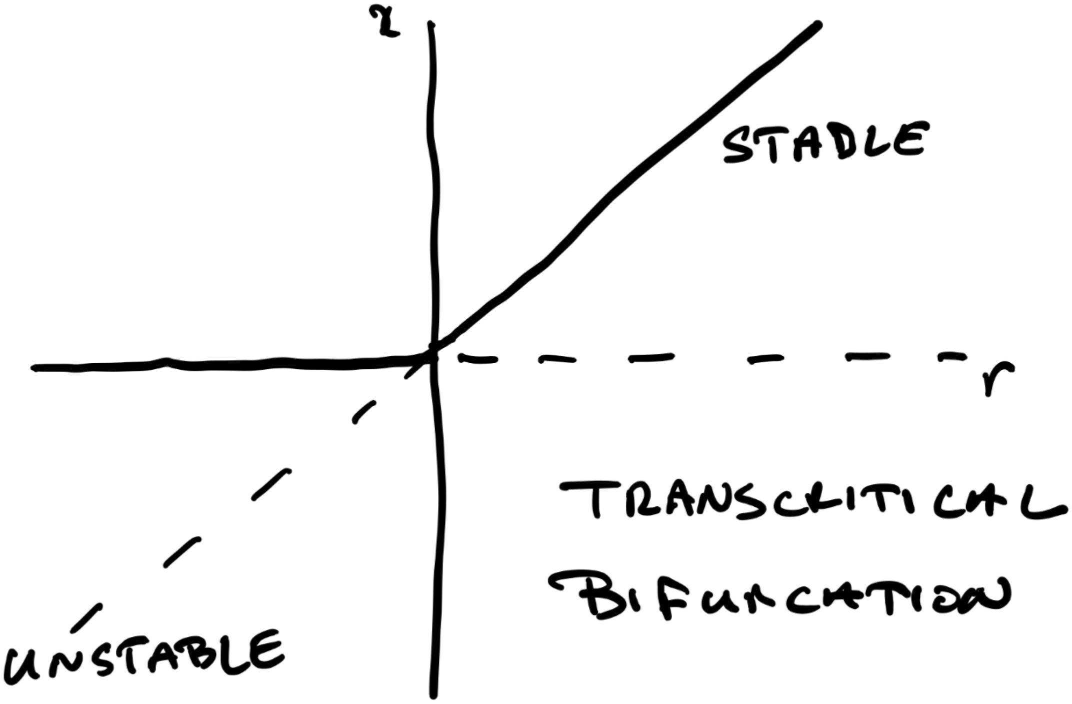
For (4), the opposite occurs as fixed points appear out of nowhere as r increases from $r < 0$. Try it as an exercise.

2.3.2 Transcritical bifurcation

Up next is the transcritical bifurcation whose normal form is

$$\frac{dx}{dt} = rx - x^2. \tag{5}$$

For the transcritical bifurcation, one fixed point moves along the x-axis and exchanges stability with the other at $x^* = 0$ when they meet. Specifically, by considering $rx - x^2 = 0$, we find that the fixed points are $x^* = 0$ and $x^* = r$. We also have that $f'(r) = -r$ and $f'(0) = r$. Thus, when $r < 0$, $x^* = r$ is unstable and $x^* = 0$ is stable. The bifurcation occurs when $r = 0$ and the stability of the two fixed points changes as $x^* = 0$ is unstable and $x^* = r$ is stable when $r > 0$. This is depicted below in the bifurcation diagram.

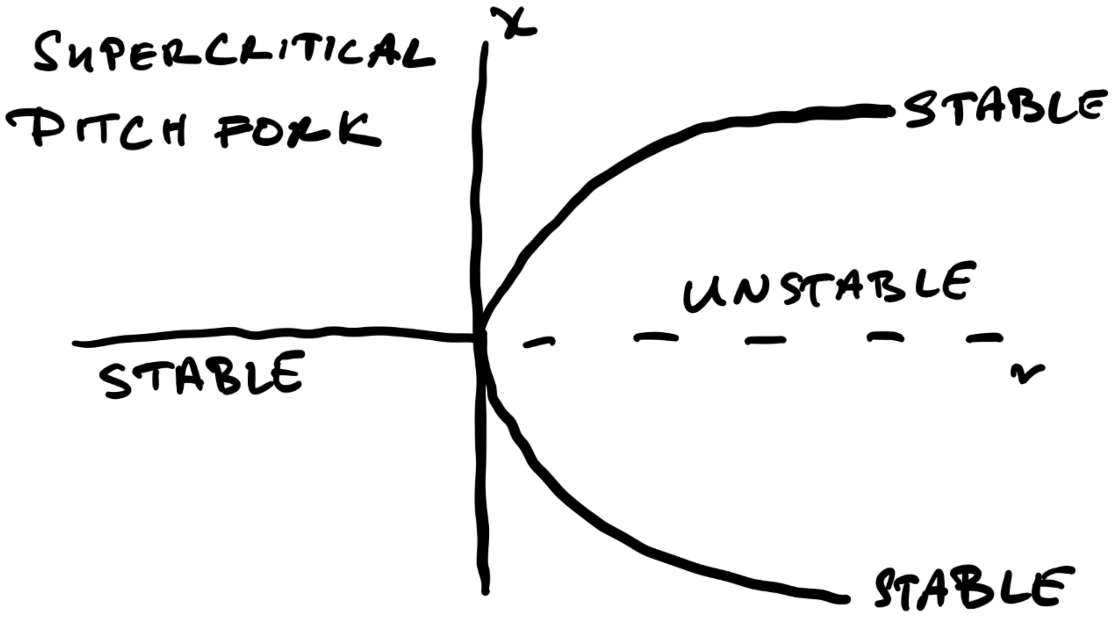


2.3.3 Pitchfork bifurcation

The last type of bifurcation is called the pitchfork bifurcation and it comes in two flavours – supercritical and subcritical. The normal form of the supercritical pitchfork is the differential equation

$$\frac{dx}{dt} = rx - x^3. \tag{6}$$

Here, we see that for $r < 0$, $rx - x^3 = 0$ emits just a single real solution, $x^* = 0$. The bifurcation occurs at the bifurcation point $r = 0$, and when $r > 0$, we have the fixed points $x^* = 0, \pm\sqrt{r}$. By considering $f'(x) = r - 3x^2$, we see that when $r > 0$, $f'(0) > 0$ and $f'(\pm\sqrt{r}) < 0$. Thus, $x^* = 0$ is unstable, while $x^* = \pm\sqrt{r}$ is stable. Again, the supercritical pitchfork bifurcation can be described through a bifurcation diagram

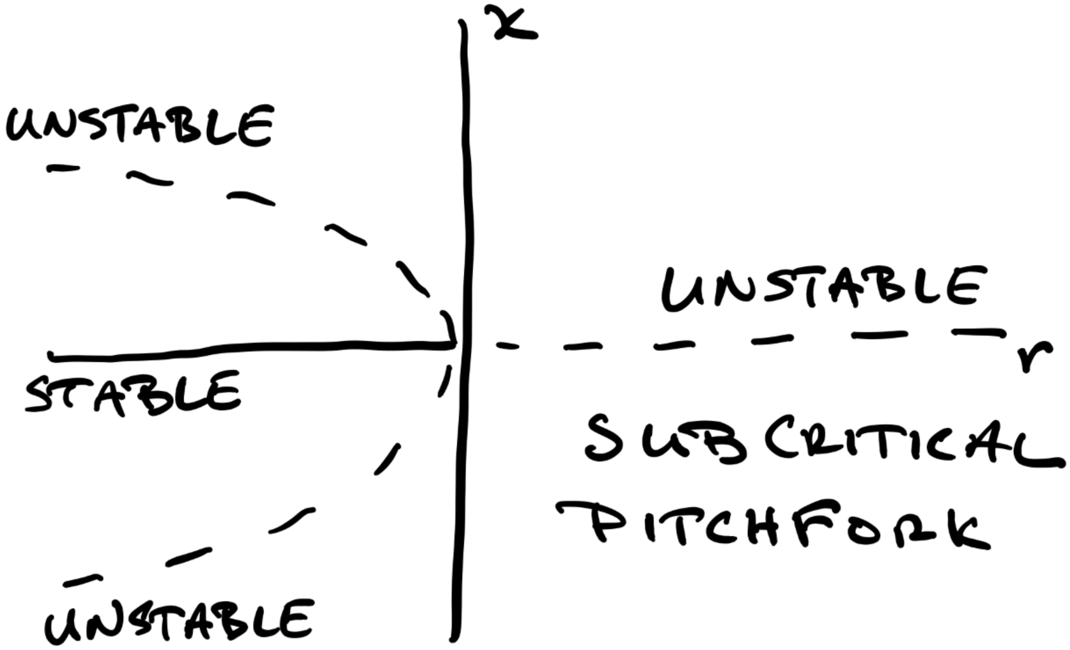


From the bifurcation diagram, you can clearly see the reason for referring to this as a pitchfork. You can also notice the symmetry. This is not a coincidence. Going back to (6), we see that the substituting $x \to -x$ does not change the equation – it is invariant under this transformation. Thus, we see that there is a left-right symmetry in the equations that is preserved as r is varied through the bifurcation point.

The other flavour is the subcritical pitchfork bifurcation whose normal form is

$$\frac{dx}{dt} = rx + x^3. \tag{7}$$

Without boring you too much with the details, for this case, we have that when $x^* = 0$ is the only fixed point, $r > 0$ and, if fact, it is unstable! As r decrease through the bifurcation point at $r = 0$, we have the familiar three fixed points $x^* = 0, \pm\sqrt{r}$, but now their stabilities are switched – $x^* = 0$ is stable, while $x^* = \pm\sqrt{r}$ are unstable. Putting this all together in the bifurcation diagram, we have that



There isn't much stability here, especially when $r > 0$ and all trajectories head off to $x \to \pm\infty$. It is important to remember that the normal forms are meant only to give a picture of what occurs *near* the bifurcation point. For a general $f(x)$ which exhibits a subcritical pitchfork bifurcation (or other bifurcations for that matter!), far away from the bifurcation point, additional, higher-order terms in $f(x)$ may become important and reveal other fixed points that may be stabilising. By finding fixed points, assessing their stability, and analysing their bifurcations, we can reveal how interesting trajectories can emerge by allow parameters to vary.

2.4 A final comment: numerical methods and asymptotic solutions

Indispensible tools, especially for nonlinear systems, are numerical methods for generating approximate solutions to initial value problems. Often, they are the most direct way to generate solutions to these problems. These methods, also referred to as time integration schemes, take an initial condition and march the solution forward in time using discrete increments, Δt. Generally speaking, there are two types – multi-stage schemes (e.g. Runge-Kutta) and multi-step schemes (e.g. Adams methods). Methods are characterised by their *order of accuracy*, the power of Δt for which the numerical solution converges to the true solution, and their *stability* which, together with accuracy ensure convergence. We will not go into the details of these methods, but the Jupyter notebooks associated with the course will rely on these methods when generating numerical solutions, so we mention them here.

The primary disadvantage of numerical schemes is that they can generate a solution only for a single set of parameters at a time. Often, especially for biological problems, one would like to explore the dependence of the solution on the parameters that appear in the model, but sweeping the parameter space with a numerical method can be time consuming. This is one reason that numerical methods cannot be relied on exclusively and analytical techniques, such as linear stability analysis, remain important. Another class of analytical methods are asymptotic analyses aimed at obtaining approximate solutions in the limit where parameters appearing in the equation are vanishingly small. By considering an expansion of the solution in a series involving the small parameter, the model equations often simplify. We'll use asymptotic techniques below to reduce a two-dimensional system to two, one-dimensional systems.

3 One dimensional systems in biology

Armed with the tools described above we can begin to analyse the mathematical models of biological processes that are one dimensional systems.

3.1 Logistic growth

One of the most classical applications of mathematics in biology is population dynamics, where the model is intended to predict the growth or decay of a population of organisms (we'll get to which ones in a bit) over time.

3.1.1 Exponential growth

First, let's imagine that the population size is given by the smooth function of time, $x(t)$. We assume in describing its dynamics that the population is already sufficiently large that stochastic fluctuations in organism numbers are small relative to the size of the population. Additionally, we assume that changes in the population size can only occur through birth and death. This is known as a closed population as migration (another mechanism for a population to change its size) to and from other populations is prohibited. Thus, knowing the birth and death rates, we can obtain an equation for the population density.

In formulating the birth and death rates, we need to make a few assumptions, the first of which will be that the rate of birth is proportional to the number of organisms. Based on this, if the population size at time t is $x(t)$, then the number of new births that occur over from time t to $t+\tau$ is approximately $bx\tau$, where $b \geq 0$ is the (constant) per capita birth rate. Similarly, we will assume that the number of deaths is also proportional to $x(t)$ and therefore the number of deaths that occur from time t to $t+\tau$ is approximately $\mu x\tau$, where $\mu \geq 0$ is the (constant) per capita death rate. Putting this all together, we have that

$$x(t+\tau) - x(t) = (b-\mu)\tau x + O(\tau^2).$$

Dividing through by τ and taking the limit as $\tau \to 0$, we obtain the familiar linear differential equation

$$\frac{dx}{dt} = rx. \tag{8}$$

where $r = b - \mu$ is the net growth rate. For the initial condition $x(0) = x$, the solution is

$$x(t) = x_0 e^{rt}$$

and we obtain exponential growth if $r > 0$ (birth rate exceeds the death rate) and exponential decay if $r < 0$ (death rate exceeds the birth rate). In terms of the population dynamics, we see that there are just two outcomes at $t \to \infty$ – either the population grows without bound, or it dies out completely, there is no middle ground!

This view was that originally observed by Malthus in 1798 who predicted that the exponential growth in population would spell disaster as the rate of food production would not be able to keep pace with population growth. Given the gravity of what this model predicts, questioning its validity is warranted, if not encouraged! Are the constant birth and death rates per capita reasonable assumptions? Have we taken into account all necessary factors?

3.1.2 Logistic growth

It turns out that while populations are often seen to grow exponentially at first, the growth rate changes over time and is seen to decrease as the population size increases. The simplest assumption that one can make is that the growth rate decreases linearly with the population size $r(1 - x/K)$ where $r > 0$ is the growth rate for very small populations ($x \approx 0$) and $K > 0$ is the population size at which the growth rate is zero. In this context, r is referred to as the *intrinsic growth rate*, a name which should be evident from our previous statement. The parameter K is called the *carrying capacity* and we will see why shortly!

Replacing the growth rate in (8) by $r(1-x/K)$, we arrive at the **logistic equation** (with associated initial condition)

$$\begin{aligned} \frac{dx}{dt} &= rx\left(1 - \frac{x}{K}\right). \\ x(0) &= x_0 \end{aligned} \tag{9}$$

attributed to Verhulst who introduced it in 1838. This change to logistic growth yields a nonlinear differential equation with a quadratic nonlinearity. Despite this, an analytical solution is possible. Perhaps this is why it is such a fundamental model!

Applying separation of variables to (9), we obtain

$$\int_{x_0}^{x} \frac{dy}{y(K-y)} = \int_0^t \frac{r}{K} d\tau. \tag{10}$$

Using partial fractions to write

$$\frac{1}{y(K-y)} = \frac{1}{K}\left(\frac{1}{y} + \frac{1}{K-y}\right),$$

the integrals above yield

$$\frac{1}{K}\left(\log x - \log(K-x)\right) - \frac{1}{K}\left(\log x_0 - \log(K-x_0)\right) = \frac{r}{K}t.$$

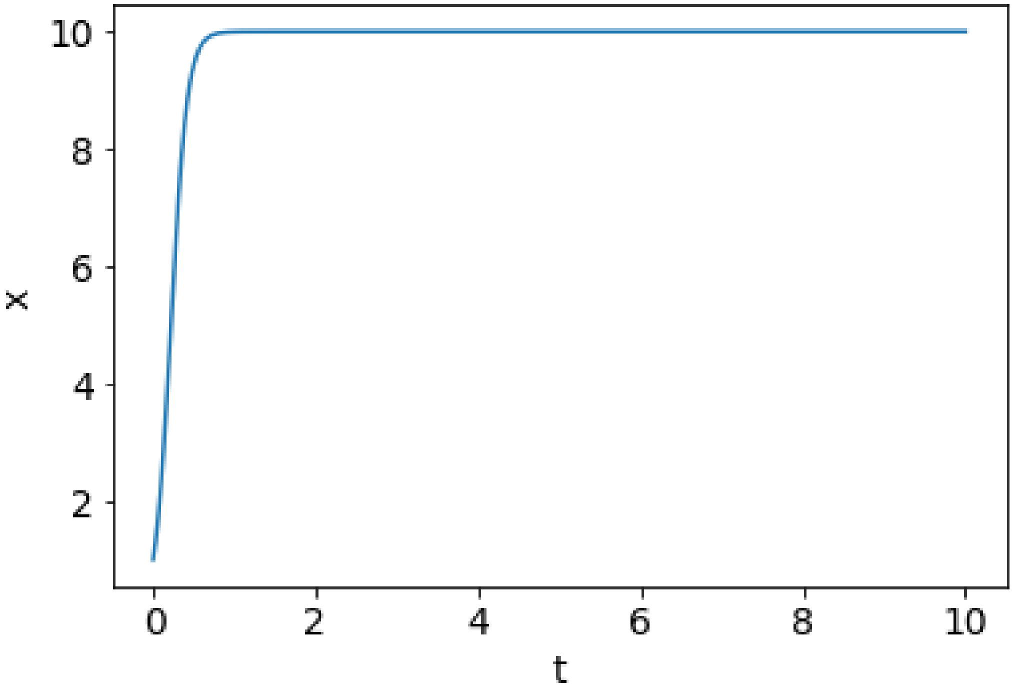
After some algebraic manipulation to solve of x, we arrive at the solution

$$x(t) = \frac{Kx_0}{x_0 + (K-x_0)e^{-rt}} \tag{11}$$

Solutions can be plotted using the code here.

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x0 = 1.0
r = 10.0
K = 10.0
t = np.linspace(0,10,1000)
x = K*x0/(x0 + (K - x0)*np.exp(-r*t))

plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("t")
plt.ylabel("x")
plt.plot(t,x);
```



Let's look more closely now at what has changed from our picture before where we had exponential growth. Recalling that r is a positive constant, we see that $x \to K$ as $t \to \infty$. Now we see why K is called the carrying capacity. It refers to the population size that can be maintained by the available resources. This is illustrated through by following problem:

Suppose that the growth rate of a population is $r = aC$, where $C(t)$ is the concentration of nutrients. Suppose that at $C(0) = C_0$ and that $dC/dt = -(1/b)dx/dt$, i.e. consumption of a unit of nutrient produces b units of population size. Show that x obeys the logistic equation. What are the expressions for r and K?

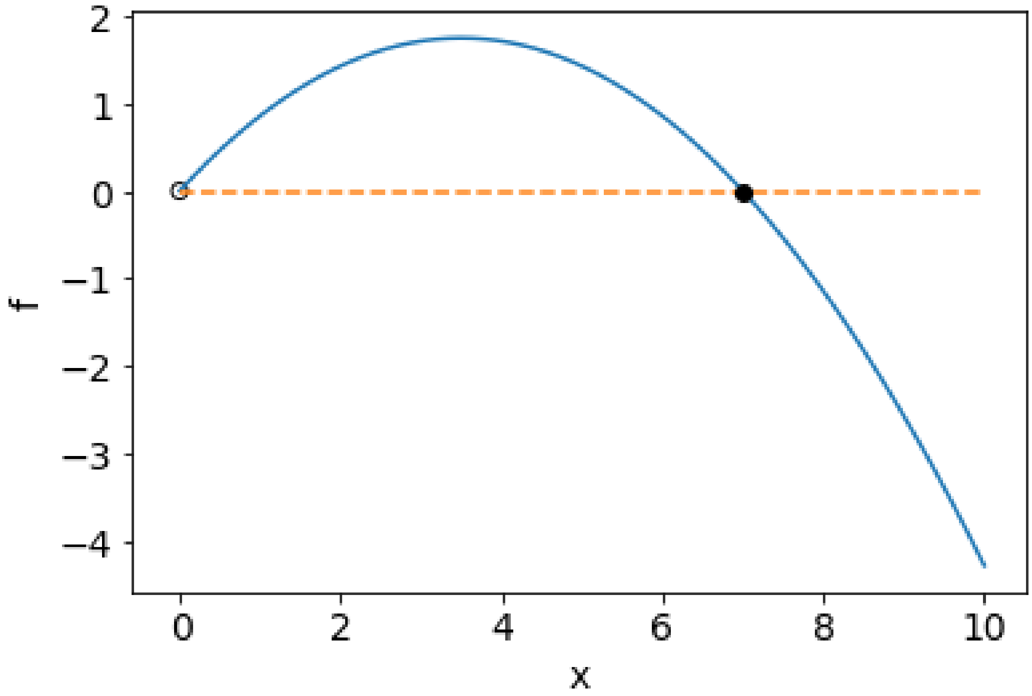
At this point, its worth revisiting what we mean by ‘organism.’ As you might imagine, such a simple equation would capture what happens with very simple organisms. These include micro-organisms that reproduce by dividing and rely on nutrients that are more or less distributed uniformly throughout the population. The population size of other, more complex organisms, such as plants, animals and people, will likely depend on many more factors such as competition, predation and mutualism, and a more refined breakdown of the population, for example the age structure. Nevertheless, we will see that logistic growth is an important ingredient in many models arising in mathematical biology.

Fixed points It’s worth noting that despite being a nonlinear equation, we were able to obtain a solution to the logistic equation via separation of variables. Despite having the solution, it’s worth pretending that we don’t explore what happens utilising some of the analytical techniques for nonlinear one-dimensional systems outlined above.

Let’s start off by finding the fixed points of (9). Considering $f(x) = rx(1 - x/K) = 0$, one finds fixed points $x^* = 0$ and $x^* = K$. Given the quadratic nature of $f(x)$ and since $r > 0$, we see that $f > 0$ for $0 < x < K$ and $f < 0$ for $x > K$. Thus, $x^* = 0$ is unstable, while $x^* = K$ is stable. This can be shown graphically using the code below

[2]:

```python
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

r = 1.0
K = 7.0
x = np.linspace(0,10,1000)
f = r*x*(1 - x/K)

plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("x")
plt.ylabel("f")
plt.plot(x,f);
plt.plot(x,0*f,'--')
plt.plot(K,0,'ko');
plt.scatter(0,0, edgecolor="black", color="none");
```



I hope that the usefulness of the tools described earlier on is somewhat clear. We were able to ascertain that as $t \to \infty$, $x \to K$ without ever having to find the solution to the differential equation and subsequently taking the limit as $t \to \infty$. As you might imagine, this becomes even more useful when there is no possibility of finding a solution!

Mathematical Biology - Week 2

October 10, 2022

0.0.1 The logistic equation in epidemiology: SIS model

For many diseases, such as meningitis, plague, malaria, and venerial diseases, populations can be divided into two types – those that are susceptible and those that are infective. This is in contrast to viral diseases where there is a third category of the population that has immunity, but more on that later. Here, we call the size of the susceptible population, S, and the infective population, I and the total population size, N, where $S + I = N$. We assume that the birth and death rates are equal and both given by δ and that all newborns are susceptible. While this means that N is constant and the rate of change of S and I due to deaths is $-\delta S$ and $-\delta I$, respectively, while births only affects S with the rate of change, δN. Additionally, the rate at which the susceptible population becomes infective is $-\lambda SI/N$, while the rate at which the infective population returns to be susceptible due to recovery is $-\gamma I$. Putting everything together, we have

$$\frac{dS}{dt} = -\lambda SI/N + \gamma I + \delta N - \delta S, \tag{1}$$

$$\frac{dI}{dt} = \lambda SI/N - \gamma I - \delta I, \tag{2}$$

$$N = S + I. \tag{3}$$

This is called the **SIS model**. Notice that the terms $-\lambda SI/N$ and γI in the equation for S appear in the equation for I, but with the opposite sign. This reflects the fact that those that are no longer infective must become susceptible, and similarly, those that were susceptible become infective.

Something else to notice is that this does not appear to be a one dimensional system as there are two differential equations, one for I and one for S. However, by virtue of the fact that N is constant, we know once we know I, we can immediately find S through $S = N - I$. Using this fact, the differential equation for I becomes,

$$\frac{dI}{dt} = \lambda(1 - I/N)I - \gamma I - \delta I, \tag{4}$$

which can be written as

$$\frac{dI}{dt} = (\lambda - \gamma - \delta)I\left(1 - \frac{\lambda}{N(\lambda - \gamma - \delta)}I\right), \tag{5}$$

and thus recover the logistic equation with $r = \lambda - \gamma - \delta$ and $K = N(\lambda - \gamma - \delta)/\lambda$. Knowing what you know about the solution to the logistic equation and what the parameters λ, γ, δ and N represent, what does the SIS model predict about the infective population?

0.0.2 Logistic growth with harvesting

Recall that one of the many assumptions made is that our system is closed – the population changes solely as a result of birth or death. We can relax this assumption through a simple modification of the logistic equation that accounts for *constant-rate harvesting.* This modified logistic equation is given by

$$\frac{dx}{dt} = rx\left(1 - \frac{x}{K}\right) - H \tag{6}$$

where H is the harvesting rate. We find the fixed points of (6) by considering $f(x) = rx(1-x/K) - H = 0$. The solutions to this quadratic equation are

$$x^*_\pm = \frac{K \pm \sqrt{K^2 - 4HK/r}}{2}. \tag{7}$$

For $H > rK/4$, the roots are complex and hence no fixed points, though there is still an outcome, just a catastrophic one! For this case $f(x) < 0$ for all x, and hence we will reach $x = 0$ in a finite time – the population will die out. This is certainly catastrophic for the population, however, the catastrophe that I am referring to is the mathematical one. We see that if we start at $H < rK/4$ and slowly increase it, we will have a stable fixed point that approaches $x^* = K/2$. The moment H goes above $rK/4$, the fixed point disappears and we have the solution $x = 0$ at long time. This discontinuous jump in the long time solution is known as a catastrophe.

Provided $H \leq rK/4$, there are two real roots and hence two fixed points with x^*_+ being asymptotically stable while x^*_- is unstable. This can be explained using linear stability analysis, or by examining for which values of x is $rx(1 - x/K) - H$ is positive or negative.

Hopefully, from our discussion above, you can see that saddle-node bifurcation has occurred at the bifurcation point $H = rK/4$, where we went from having two fixed points to suddenly having none. All that has been discussed here is neatly summarised in the bifurcation diagram below with the added arrows indicating the flow for fixed H.

0.1 Insect outbreak: the spruce budworm

One cannot take a mathematical biology course without encountering the spruce budworm. It shows how a simple, one-dimensional systems can yield rich dynamics with relevant predictions of a biological phenomenon. The spruce budworm is an insect that infects fir trees in eastern Canada and a budworm infestation can lead to complete defoliation of a forest in about 4 years. The budworm is preyed upon by birds that for low budworm population feed only upon the budworms once the budworm population has reached a certain level. Assuming logistic growth for the budworm population in the absence of birds, the budworm population size, N, is governed by the differential equation

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - p(N), \tag{8}$$

where $p(N)$ is the rate of change in the budworm population size due to predation by birds. The predation can be described by the function $p(N) = BN^2/(A^2 + N^2)$ where $B > 0$ is the predation rate as $N \to \infty$ and $A > 0$ provides a measure of the threshold population size where predation suddenly increases. This leaves us with

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - \frac{BN^2}{A^2 + N^2}. \tag{9}$$

0.1.1 Nondimensionalisation

Thus far we've dealt directly with the dimensional form of the differential equation, meaning that the parameters in the equations have relevant dimensions associated with them. While this is perhaps convenient for making direct predictions from measured values of the parameters, working with dimensional parameters can at best keep the equations looking 'un-tidy,' and at worst obscure the mathematical structure of the equations or relevant approximations that can be made.

Here, we nondimensionalise N by A, such that, $N = Ax$. Thus, the nondimensional budworm population size is given by x. For time, we take $t = (A/B)\tau$, so τ is the nondimensional time. Substituting these expressions into (9) for N and t gives

$$B\frac{dx}{d\tau} = rAx\left(1 - \frac{Ax}{K}\right) - \frac{Bx^2}{1 + x^2}. \tag{10}$$

Dividing both sides by B and defining $R = rA/B$ and $k = K/A$, we have

$$\frac{dx}{d\tau} = Rx\left(1 - \frac{x}{k}\right) - \frac{x^2}{1 + x^2}. \tag{11}$$

We see that nondimensionalisation has done a couple of things. The first is that it has 'tidied up' our equation – we went from four parameters to two. Additionally, all of the parameters are found in the logistic growth term on the right hand side. This will prove useful when we go to find the fixed points.

0.1.2 Fixed points

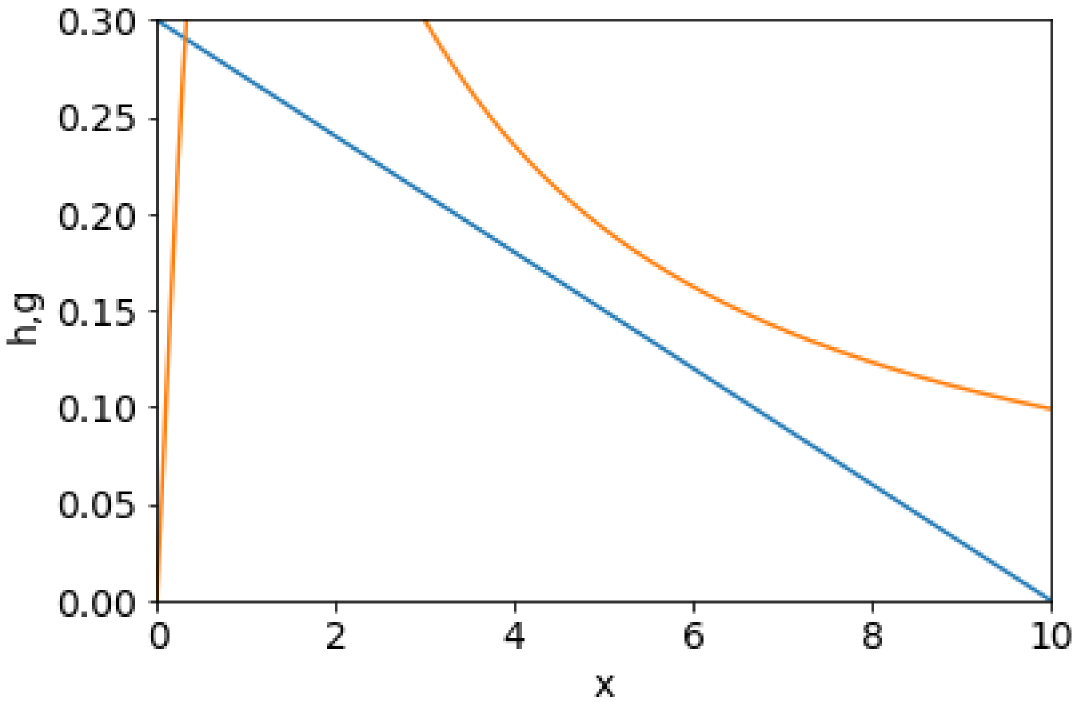
We first note that by inspection of (11), $x = 0$ is a fixed point. To find the other fixed points, we consider

$$R\left(1 - \frac{x}{k}\right) = \frac{x}{1 + x^2} \tag{12}$$

where the logistic growth is balanced by the predation. Thus, the fixed points occur when the line given by the left-hand side of (12) intersects the curve given by the right-hand side. Naturally, we'd like to explore how these fixed points vary with R and k and, in doing so, we see the benefit of our choice of nondimensionalisation. The curve given by $h(x) = x/(1 + x^2)$ does not depend on either parameter, and as for the line $g(x) = R(1 - x/k)$, R is the y-intercept, while k is the where it crosses the x-axis. This makes graphical evaluation quite easy!

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

R = 0.3
k = 10.0
x = np.linspace(0,10,1000)
h = R*(1 - x/k)
g = x/(1 + x**2);

plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("x")
plt.ylabel("h,g")
plt.plot(x,h)
plt.plot(x,g)
plt.ylim(0,R)
plt.xlim(0,k);
```



The code above can be used to plot the curve $h(x)$ and the line $g(x)$ for different R and k. For values of k sufficiently low, there will be a single additional fixed point for $R > 0$. When k is large, however, we can have either one, two, or three additional fixed points. Try it! Consider the case of three additional fixed points, labeled a, b and c, such that $a < b < c$. If we fix the value of k and decrease R, we see that b and c collide and we go from three to one additional fixed point via a saddle-node bifurcation. Similarly, by increasing R, a and b collide and we'll have a saddle-node bifurcation there as well.

0.1.3 Stability

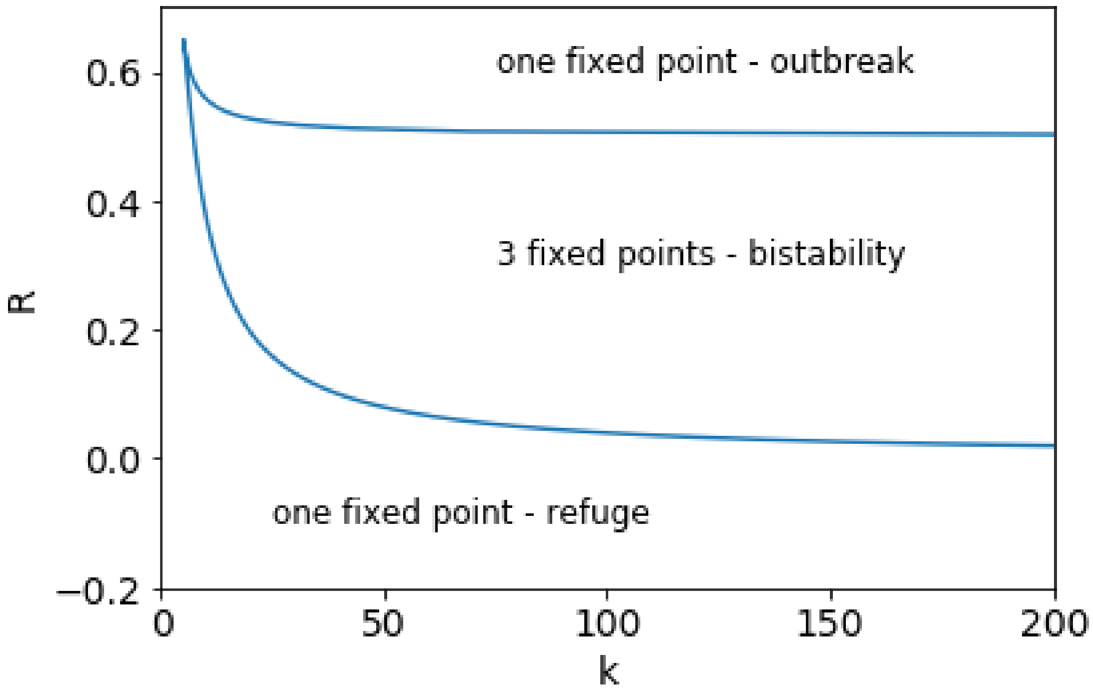
We know that $x^* = 0$ is a fixed point, and since $f(x) > 0$ just to the right of $x^* = 0$, this is an unstable fixed point. Now, turning our attention to the case where we have three additional fixed points, a, b, and c. Since the stability of the fixed points must alternate as x increases from 0 (this is due to the continuity of f), a and c will be stable, while b will be unstable.

At this point, it's important to remember that this is a model of a biological process and the fixed points that we found have an interpretation in the context of the system. The stable fixed points $x^* = a$ and $x^* = c$ correspond to the *refuge* and *outbreak* population levels, respectively. The unstable fixed point $x^* = b$ is referred to as the *threshold* as initial populations $x_0 > b$ increase to the outbreak level, while for $x_0 < b$ the population will decrease to the refuge size. We notice something important here – the system exhibits bistability. For a fixed set of parameters, we can have two stable fixed points.

We can find the regions in (k, R) parameter space where there is only 1 additional fixed point, or where there are 3 by finding parametric curves $R(x)$ and $k(x)$ that give the bifurcation curves – the curves containing the bifurcation points. These are obtained by considering simultaneously $f(x) = 0$ and $f'(x) = 0$, which state that the bifurcation point correspond to a fixed point, and it occurs where the tangent to the curve is zero, i.e. where the line given by $R(1 - x/k)$ is tangent to the curve $x/(1 + x^2)$. Performing this calculation, we find that $R(x) = 2x^3/(1 + x^2)^2$ and $k(x) = 2x^3/(x^2 - 1)$. The code below plots these curves and indicates the number of fixed points in each region.

[2]:

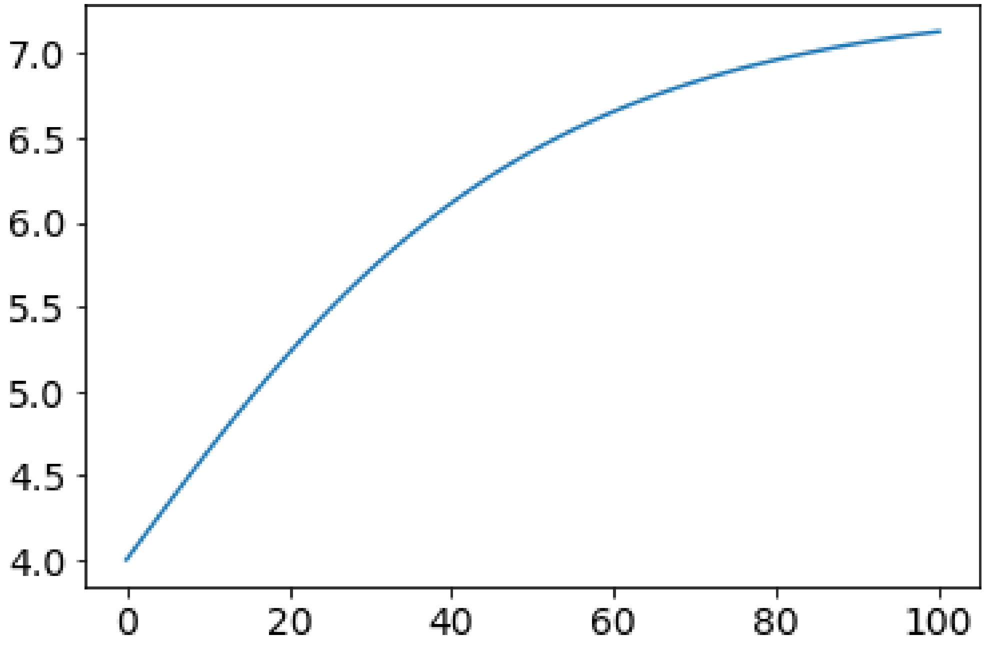
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.linspace(1.005,100,10000)
R = 2*x**3/(1 + x**2)**2;
k = 2*x**3/(x**2 - 1)

plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("k")
plt.ylabel("R")
plt.plot(k,R);
plt.ylim(-0.2,0.7)
plt.xlim(0,200);
plt.text(75, 0.6, "one fixed point - outbreak", fontsize=12)
plt.text(75, 0.3, '3 fixed points - bistability', fontsize=12)
plt.text(25, -0.1, 'one fixed point - refuge', fontsize=12);
```



How did we do with our analysis: use the code below to integrate the equations numerically and plot the solution as a function of time. If you vary the parameters, does the solution go to where you think it should?

[3]:

```
from scipy.integrate import odeint

def dx_dt(x, t):
    R = 0.5
    k = 10.0
    f = R*(1 - x/k)
    g = x/(1 + x**2);
    return f-g

t = np.linspace(0,100,100)
x0 = 4.0  # the initial condition
x = odeint(dx_dt, x0, t)
x = np.array(x).flatten()

plt.plot(t,x);
```



One important feature of this system is that it can exhibit *hysteresis* meaning that varying the parameters in such a way that they return to their initial values does not return the system to its original state.

For example, suppose that the parameters R and k are such that we have four fixed points (including $x^* = 0$). Let's suppose that the population is at the refuge size, $x = a$. Now, imagine that R is increased and the saddle-node bifurcation occurs making $x^* = c$ the only stable fixed point. The system will move to the outbreak size. Even if we reduce R to its original value, we will be above the threshold and continue to move to c rather than a. Thus, once the outbreak occurs, reducing R to its original value won't solve the problem. How might you go about returning the population of the refuge value?

0.2 Chemical kinetics

While most of what we have discussed so far describes biological processes at the scale of populations of many organisms, similar equations arise for basic chemical processes that occur at the molecular level. Additionally, the connection with a chemical process provides a more systematic way of producing a mathematical model given specific chemical reaction.

Suppose we have two molecules, A and B, that react to form another molecule C at rate k_1. This is expressed symbolically as

$$A + B \xrightarrow{k_1} C. \tag{13}$$

Writing the concentration of the molecules as $a = [A]$, $b = [B]$, and $c = [C]$, the *law of mass action* tells us that the reaction rate is proportional to the product of the concentration of the reactants.

In this case, we'll have

$$\frac{dc}{dt} = k_1 ab \tag{14}$$

and the depletion rates of a and b are

$$\frac{da}{dt} = -k_1 ab, \frac{db}{dt} = -k_1 ab. \tag{15}$$

Now, suppose instead, the reaction is such that one A combines with two B to form one C with a rate k_1,

$$A + 2B \xrightarrow{k_1} C. \tag{16}$$

In this case, the law of mass action states that dynamics of the concentrations are

$$\frac{dc}{dt} = k_1 ab^2, \tag{17}$$

$$\frac{da}{dt} = -k_1 ab^2, \tag{18}$$

$$\frac{db}{dt} = -2k_1 ab^2. \tag{19}$$

The coefficients (1,2,1) in front of A, B, and C in (16) are referred to as *stoichiometric coefficients.* If this reaction is reversible with rate k_2 such that

$$C \xrightarrow{k_2} A + 2B, \tag{20}$$

the differential equations for the concentrations are

$$\frac{dc}{dt} = k_1 ab^2 - k_2 c, \frac{da}{dt} = k_2 c - k_1 ab^2, \frac{db}{dt} = 2k_2 c - 2k_1 ab^2. \tag{21}$$

Hopefully, from this simple example, you've noticed that there is a process for going from the symbolic representation of the chemical reactions to the differential equations for the dynamics of the concentrations. Let's state the process more generally.

Suppose there are n molecules, or chemical species, $X_1, \ldots, X_n$ with concentrations, $x_1, \ldots, x_n$, respectively. Suppose also that the molecules can undergo r reactions, each of which may be expressed as

$$s^R_{1j} X_1 + s^R_{2j} X_2 + \cdots + s^R_{nj} X_n \xrightarrow{k_j} s^P_{1j} X_1 + s^P_{2j} X_2 + \cdots + s^P_{nj} X_n \tag{22}$$

for $j = 1, \dots, r$, where s_{ij}^R and s_{ij}^P and are the stoichiometric coefficients on the reactant and product sides, respectively for species i and reaction j. Writing $s_{ij} = s_{ij}^P - s_{ij}^R$, the law of mass action gives the differential equation for the concentration x_i as

$$\frac{dx_i}{dt} = \sum_{j=1}^{r} s_{ij} k_j \prod_{l=1}^{n} x_l^{s^{R}lj}, \tag{23}$$

for $i = 1, \dots, n$.

0.2.1 Enzyme dynamics: Michaelis-Menten model

Now that we have the general framework for obtaining differential equations from chemical reactions, let's consider an enzymatic reaction. The biochemical reactions that are taking place within organisms often involve proteins called *enzymes* that serve as catalysts to efficiently facilitate the reaction. The enzymes react selectively on compounds that are referred to as *substrates*. Haemoglobin on red blood cells is an example of an enzyme that combines with its substrate, oxygen. Enzymes are important as they often serve as activators or inhibitors of reactions that govern fundamental biological process, such as metabolism, the process by which energy is made available within cells.

The most basic model enzymatic reaction is the process proposed by Michaelis and Menten (1913), involving substrate S and enzyme E, that combine to form complex SE which, in turn is converted to product, P, releasing the enzyme. This reaction is described by

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} SE \xrightarrow{k_2} P + E \tag{24}$$

Applying mass action to the process described above and calling $c = [SE]$, the differential equations for the concentrations are

$$\frac{dc}{dt} = k_1 se - (k_{-1} + k_2)c, \tag{25}$$

$$\frac{de}{dt} = -k_1 se + (k_{-1} + k_2)c, \tag{26}$$

$$\frac{ds}{dt} = -k_1 se + k_{-1}c, \tag{27}$$

$$\frac{dp}{dt} = k_2 c. \tag{28}$$

From this set of equations, we notice a couple of things. The first is that $d(c + e)/dt = 0$, and therefore, if at $t = 0$, $e(0) = e_0$ and $c(0) = 0$, we have $c + e = e_0$. The second thing to notice is that since the equation for p only depends on c, we have

$$p(t) - p(0) = k_2 \int_0^t c(\tau) d\tau. \tag{29}$$

Thus, once we know c, p immediately follows through integration – there's nothing to solve for! Therefore, we need only consider the two coupled differential equations

$$\frac{dc}{dt} = k_1 e_0 s - (k_1 s + k_{-1} + k_2)c, \tag{30}$$

$$\frac{ds}{dt} = -k_1 e_0 s + (k_1 s + k_{-1})c \tag{31}$$

with the initial condition $s(0) = s_0$ and recall $c(0) = 0$.

0.2.2 Nondimensionalisation

We nondimensionalise this system of differential equations as follows

$$t = \tau/(k_1 e_0), s = s_0 u, c = e_0 v \tag{32}$$

to become

$$\epsilon \frac{dv}{d\tau} = u - (u + K)v, \tag{33}$$

$$\frac{du}{d\tau} = -u + (u + K - \lambda)v \tag{34}$$

where $\epsilon = e_0/s_0$, $K = (k_{-1} + k_2)/(k_1 s_0) = K_m/s_0$ and $\lambda = k_2/(k_1 s_0)$. The initial conditions for the nondimensional system are $u(0) = 1$ and $v(0) = 0$.

0.2.3 Quasi-steady approximation

Now, you must be thinking – this is not a one-dimensional system, the dynamics are governed by two ODEs! This is most certainly correct, but the reason that it is appearing now is that the nondimensionalisation has revealed that a simplifying assumption might be possible. If the initial concentration of enzyme is small relative to that of the substrate, we have that $\epsilon \ll 1$ and assuming the remaining coefficients are $O(1)$, we can take $\epsilon = 0$ and the system becomes

$$0 = u - (u + K)v, \tag{35}$$

$$\frac{du}{d\tau} = -u + (u + K - \lambda)v. \tag{36}$$

We can immediately solve the algebraic equation to give $v = u/(u + K)$ and upon substituting this into the differential equation, we have the one-dimensional system

$$\frac{du}{d\tau} = -\lambda \frac{u}{u + K}. \tag{37}$$

Examining the right-hand side, we see that $f = -\lambda u/(u+K) < 0$ and thus u monotonically decreasing. We also see that $u = 0$ is an asymptotically stable fixed point.
The solution can be obtained via separation of variables,

$$\int_1^u \left(1 + \frac{K}{u'}\right) du' = -\lambda\tau. \tag{38}$$

to obtain

$$u + K \log u = 1 - \lambda\tau. \tag{39}$$

One final thing to consider is the product itself. If we consider the equation for p and nondimesionalise such that $p = s_0 w$, we obtain

$$\frac{dw}{d\tau} = \lambda \frac{u}{u+K}. \tag{40}$$

which tells us the product increases monotonically in time with a rate that decreases as the substrate u is depleted.

As you can see, the approximation $\epsilon = 0$ can be quite useful, but have a look at our expression $v = u/(u+K)$. If we take $\tau = 0$, we have $v(0) = (1+K)^{-1} \neq 0$ – we haven't satisfied the initial condition for v. What went wrong? Looking more carefully, we see that in setting $\epsilon = 0$, we removed the dynamics of v and insisted it must reach its quasi-steady state instantaneously. In reality, we won't have $\epsilon = 0$, but rather a tiny, finite number, so there will be some dynamics but they will happen very quickly. To capture this, we must instead consider the limit $\epsilon \to 0$ and perform a more involved asymptotic analysis of the problem.

Mathematical Biology - Week 3

October 18, 2022

1 Multi-dimensional systems

Thus far, we've focused on biological processes or populations dynamics that can be described by a single first-order differential equation. There are, of course, many scenarios which don't fit into this category as one must often consider a system of two or more coupled differential equations. In fact, we've already encountered this in our discussion of enzyme reaction kinetics and only through an approximation were we able to reduce the system to a single first-order differential equation. The extension to multi-dimensional systems could also be viewed as a relaxation of certain assumptions in the modelling process. For example, when describing predation of the spruce budworm, we did not consider the dynamics of the birds that prey on them, but they themselves can die out in the absence of the budworm population. 'Putting this into' the model, will perhaps lead to more insight into the true population dynamics, but it will increase the mathematical complexity as well.

As you might imagine, the theory of one-dimensional systems can be viewed as a restricted case of the more general theory developed for multidimensional systems. You will see that many of the concepts and ideas that you have already encounted will be similar in the multidimensional case, however, as trajectories are no longer restricted to the x-axis, multidimensional systems will yield richer dynamics. As a result, we will encounter some new, interesting phenomena that also have important biological implications.

In general, an n-dimensional autonomous system can be expressed as

$$\frac{dx_i}{dt} = f_i(x_1, \ldots, x_n) \tag{1}$$

for $i = 1, \ldots, n$, which in vector form becomes

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) \tag{2}$$

where $\mathbf{x} = (x_1, \ldots x_n)^T$ and $\mathbf{f} = (f_1(\mathbf{x}), \ldots f_n(\mathbf{x}))^T$. Just as we began our discussion of one-dimensional systems, it's important to note the very general result about existence and uniqueness.

Existence and Uniqueness Theorem: If the functions $f_i(\mathbf{x})$ and their first derivative $\partial f_i/\partial x_j$ are continuous for $-\infty < x_k < \infty$ $(k = 1, \ldots, n)$ then for any initial condition

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{3}$$

there exists a unique solution of (2) and (3) for a small time interval $-\delta < t < \delta$. In particular, if $\mathbf{f}$ is bounded linearly, that is

$$\|\mathbf{f}(\mathbf{x})\| \leq c_1 \|\mathbf{x}\| + c_2 \tag{4}$$

for some positive constants, c_1 and c_2, then the solutions can be uniquely extended to all $-\infty < t < \infty$.

1.1 Fixed points and their stability

As with one-dimensional system, fixed points and their stability play an important role in multidimensional systems. For multidimensional systems, however, there are more ways fixed points can be stable or unstable, which we will explore in the case of two-dimensional systems. Let's first begin with some more rigorous definitions of stability.

1. A fixed point is said to be **stable** if for any small δ_1 there exists a δ_2 such that if $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta_2$, then the solution exist for all $t > 0$ and $\|\mathbf{x}(t) - \mathbf{x}^*\| < \delta_1$ for all $t > 0$.
2. A stable fixed point is said to be **asymptotically stable** if any solution $\mathbf{x}(t)$ with $\mathbf{x}(0)$ near $\mathbf{x}^*$ converges to $\mathbf{x}^*$ as $t \to \infty$.
3. A fixed point is **unstable** if is not stable.

We can assess stability of a fixed point $\mathbf{x}^*$ by linearising $\mathbf{f}$ about $\mathbf{x}^*$ such that

$$\mathbf{f}(\mathbf{x}) = A(\mathbf{x} - \mathbf{x}_0) + \text{h.o.t.} \tag{5}$$

where the higher order terms are $O(\|\mathbf{x} - \mathbf{x}_0\|^2)$ and A is the Jacobian matrix whose elements are

$$a_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{x}^*) \tag{6}$$

Let's now consider the system of linear differential equations

$$\frac{d\mathbf{y}}{dt} = A\mathbf{y} \tag{7}$$

where $\mathbf{y} = (y_1, \dots y_n)^T$. A solution to the system is $\mathbf{y}(t) = \boldsymbol{\xi}\mathrm{e}^{\lambda t}$, if

$$A\boldsymbol{\xi} = \lambda\boldsymbol{\xi}. \tag{8}$$

Thus, λ is an eigenvalue of A and $\boldsymbol{\xi}$ is the corresponding eigenvector. Recall that the eigenvalues of A can be found by considering $\det(\lambda I - A) = 0$ and solving the characteristic equation. If the matrix A is diagonalisable with eigenvalues λ_i and eigenvectors $\boldsymbol{\xi}_i$ $(i = 1, \dots n)$, then the general solution to (7) is

$$\mathbf{y}(t) = \sum_{i=1}^{n} c_i \boldsymbol{\xi}_i \mathrm{e}^{\lambda_i t} \tag{9}$$

where the coefficients c_i for $i = 1, \dots, n$ are obtained by applying the initial condition. We will discuss the case where the matrix is defective in the specific case of $n = 2$.

It's important to remember that the system (7) is a consequence of examining the local dynamics near the fixed point, $\mathbf{x}^*$, and as you might imagine, its properties will be related to the stability of $\mathbf{x}^*$. This is all encapsulated in the following theorem:

Stability theorem Let $\mathbf{x}^*$ be a fixed point of (2), so that (5) holds. If all the eigenvalues of the Jacobian matrix A have negative real parts, then $\mathbf{x}^*$ is asymptotically stable, and $\|\mathbf{x}(t) - \mathbf{x}^*\| \leq \text{const.}\mathrm{e}^{-\mu t} (\mu > 0)$ for all $t > 0$ if $\|\mathbf{x}(0) - \mathbf{x}^*\|$ is sufficiently small. If at least one of the eigenvalues has a positive real part, then $\mathbf{x}^*$ is unstable.

Essentially, what this theorem says is that if you find a fixed point and want to evaluate its stability, *most of the time* it is sufficient to linearise **f** about the fixed point and examine the eigenvalues of the Jacobian. If one eigenvalue has a positive real part, the fixed point is unstable. If all eigenvalues have negative real parts, then the fixed point is asymptotically stable. So, it only takes one positive eigenvalue to screw up stability.

It's useful to think about this theorem in light of what we already have learned about one-dimensional systems. There, $n = 1$ and the Jacobian is just a real number, namely df/dx. Since this is the only eigenvalue, we only needed to examine whether $df/dx > 0$ or $df/dx < 0$ to test stability. For $n \geq 2$, we can assess stability by computing the partial derivatives of **f**, but must go further and diagonalise the Jacobian matrix. Let's now examine how this plays out for $n = 2$.

1.2 Two-dimensional systems: $n = 2$

Taking the system (7) with $n = 2$, we can write the matrix A as

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \tag{10}$$

To find the eigenvalues of A, we consider

$$\det\left(\begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix}\right) = (a-\lambda)(d-\lambda) - cb = 0. \tag{11}$$

Solving the quadratic equation for λ, we obtain

$$\lambda_{1,2} = \frac{\tau \pm \sqrt{\tau^2 - 4\Delta}}{2} \tag{12}$$

where $\tau = \text{trace}(A) = a + d$ and $\Delta = \det(A) = ad - bc$. We see that both eigenvalues are real if $\tau^2 - 4\Delta > 0$ and complex if $\tau^2 - 4\Delta < 0$. Now, assuming A is diagonalisable, the general solution (9) for $n = 2$ is

$$\mathbf{y}(t) = c_1\boldsymbol{\xi}_1 \mathrm{e}^{\lambda_1 t} + c_2\boldsymbol{\xi}_2 \mathrm{e}^{\lambda_2 t}. \tag{13}$$

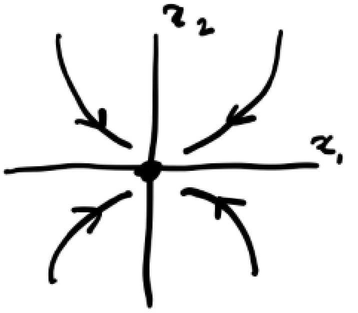
Examining the expression for the eigenvalues (12) and in light of the stability theorem, we see that we can have the following possible cases:

1. **Stable node**: $\tau^2 - 4\Delta > 0$, and $\lambda_1 < 0$, $\lambda_2 < 0$.
2. **Unstable node**: $\tau^2 - 4\Delta > 0$, and $\lambda_1 > 0$, $\lambda_2 > 0$.
3. **Saddle point**: $\tau^2 - 4\Delta > 0$, and $\lambda_1 > 0$, $\lambda_2 < 0$.
4. **Stable Star**: $\tau^2 - 4\Delta = 0$, A is diagonalisable, and $\lambda_1 = \lambda_2 < 0$.
5. **Unstable Star**: $\tau^2 - 4\Delta = 0$, A is diagonalisable, and $\lambda_1 = \lambda_2 > 0$.
6. **Stable spiral**: $\tau^2 - 4\Delta < 0$, and $\text{Re}(\lambda_1) = \text{Re}(\lambda_2) < 0$.
7. **Unstable spiral**: $\tau^2 - 4\Delta < 0$, and $\text{Re}(\lambda_1) = \text{Re}(\lambda_2) > 0$.
8. **Centre**: $\tau^2 - 4\Delta < 0$, and $\text{Re}(\lambda_1) = \text{Re}(\lambda_2) = 0$.

The different cases are illustrated graphically below.

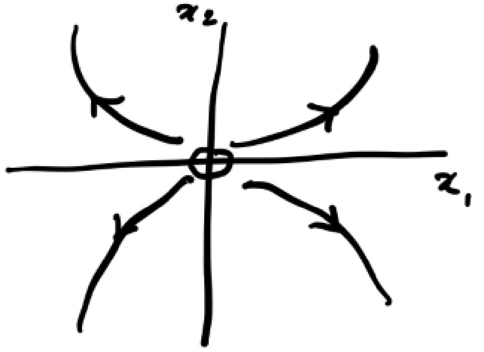
STABLE NODE

$\lambda_1 , \lambda_2 < 0$



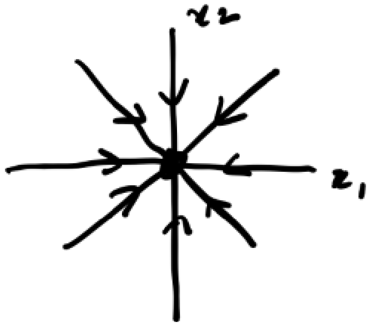
UNSTABLE NODE

$\lambda_1 \ , \ \lambda_2 > 0$



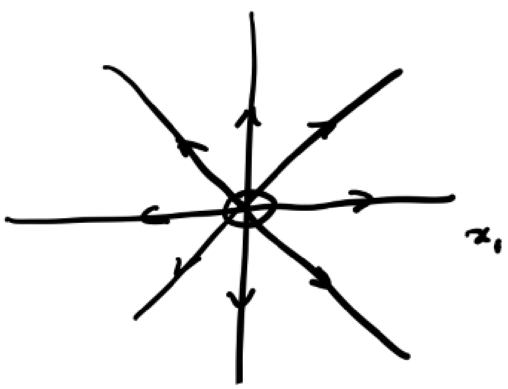
STABLE STAR

$\lambda_1 = \lambda_2 < 0$



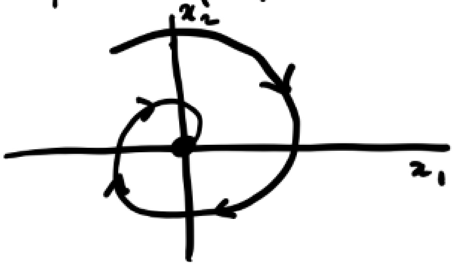
UNSTABLE STAR

$\lambda_1 = \lambda_2 > 0$



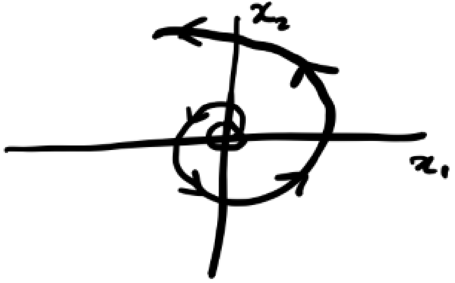
STABLE SPIRAL

$\lambda_1 = \alpha + i\beta \ , \ \alpha < 0$



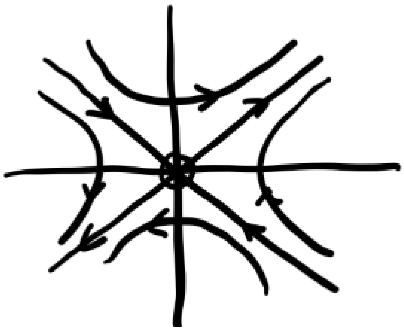
UNSTABLE SPIRAL

$\lambda_1 = \alpha + i\beta \ , \ \alpha < 0$



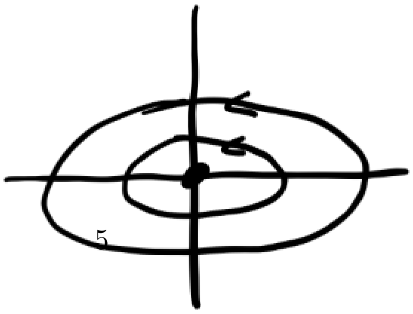
SADDLE POINT

$\lambda_1 > 0 \ , \ \lambda_2 < 0$



CENTRE

$\lambda_1 = i\beta$



We see that the freedom for the trajectories to be in a plane rather than along a line has increased the diversity of the types of fixed points that one can have. The trajectories can be along lines that head directly into a fixed point (stable node and stable star) or away from it (unstable node and unstable star), or circle around the fixed point before meeting it (stable spiral), or circle around it while moving away (unstable spiral). We also see the introduction of the saddle point, which allows for motion toward the fixed point along a particular curve called the **stable manifold** (the direction of which is described locally by the eigenvector associated with negative eigenvalue), but any perturbation away from this curve results in motion away from the fixed point, including along the **unstable manifold** emanating from the saddle, that is described locally by the eigenvector associated with positive eigenvalue. Hence, saddle points are unstable.

We also see something else that did not occur in one-dimensional systems. In the case of the centre, the trajectories move around the fixed point never getting closer, or further away. This is an example of a closed orbit, or limit cycle. We will see that in context of fully nonlinear systems closed orbits can exist and they themselves can be stable or unstable.

A final aspect to note is the case where the matrix A is not diagonalisable. In this case, the eigenvalue will be repeated, as will its associated eigenvector, or in other words, the algebraic multiplicity exceeds the geometric multiplicity. In $n = 2$, this means that an eigenvalue, λ, will be repeated, but there will be just one eigenvector, $\boldsymbol{\xi}$, associated with it. The solution is then

$$\mathbf{y}(t) = c_1\boldsymbol{\xi}\mathrm{e}^{\lambda t} + c_2(\boldsymbol{\xi}t + \boldsymbol{\zeta})\mathrm{e}^{\lambda t}, \tag{14}$$

where the vector $\boldsymbol{\zeta}$ is given by $(A - \lambda I)\boldsymbol{\zeta} = \boldsymbol{\xi}$. As the eigenvalue is repeated, we know that it must be real. Thus, if $\lambda > 0$, the fixed point is unstable, while if $\lambda < 0$, the fixed point is stable. This type of fixed point is referred to as a degenerate node and can visualised by allowing the eigenvectors of your typical node to collapse onto one another.

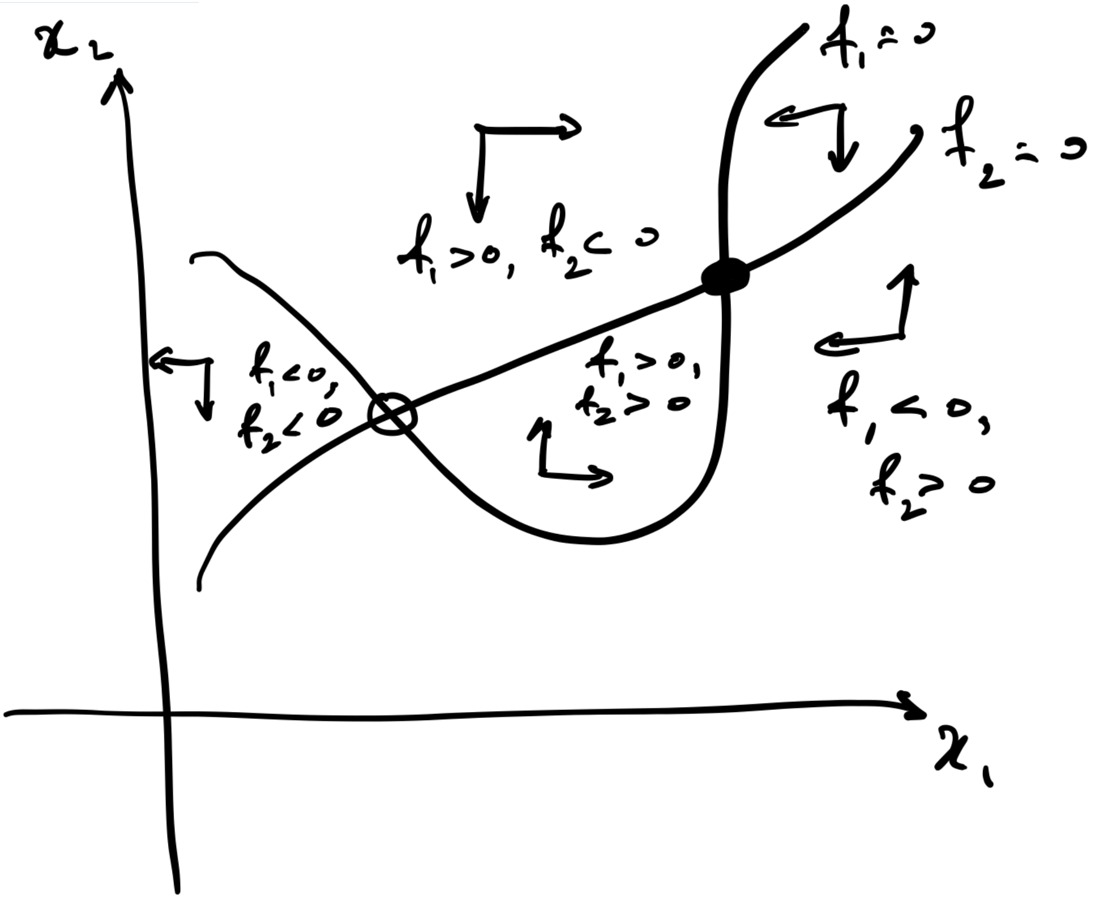
1.2.1 Phase portraits

It's important to keep in mind that the linear analysis about fixed points only gives an local description of the dynamics. To get a better picture of the global dynamics, we'd like to understand how the fixed points are connected, or if closed orbits exist. In one-dimensional systems, this we easy as we only needed to examine the sign of f between the fixed points and the phase portrait was clear. In two dimensions, the phase portrait will typically include

1. The **fixed points**.
2. The **closed orbits**. These are solutions that have the property $\mathbf{x}(t) = \mathbf{x}(t + T)$ for some $T > 0$.
3. The trajectories near fixed points and closed orbits that show the flow pattern.
4. Some indication of whether the fixed points or closed orbits are stable or unstable.

While uniqueness of the solution dictates that trajectories cannot cross, compiling the phase portrait in 2D can be challenging. Fixed points are a good place to begin when assembling the phase portrait, but to get a sense of the flow in different regions of the phase plane, one should compute the nullclines as we describe in the next section.

Nullclines The **nullclines** are the curves where either $dx_1/dt = f_1(x_1, x_2) = 0$, or $dx_2/dt = f_2(x_1, x_2) = 0$. Since the fixed points satisfy both $dx_1/dt = 0$ and $dx_2/dt = 0$, the nullclines will intersect at the fixed points. The nullclines are helpful in compiling the phase portrait as we know that when trajectories intersect the nullcline, they must be either horizontal ($dx_1/dt = 0$) or vertical ($dx_2/dt = 0$). Additionally, and perhaps more importantly, the nullclines provide the boundary between regions of phase space where $f_1 > 0$ or $f_1 < 0$, and $f_2 > 0$ or $f_2 < 0$. Thus, the nullclines give us some sense of the direction of the trajectories in different regions of the phase plane. The sketch below shows the nullclines and the regions of phase space that they establish.



1.3 Multidimensional models of biological processes

Using the theory outlined above, we will now examine several models of biological processes drawn from epidemiology, cellular biochemistry, and population dynamics that are multidimensional systems. We will see how the mathematical tools used to analyse the models lead to insight into the processes that the models describe.

1.3.1 SIS model for two interacting populations

We begin with a model from epidemiology, in fact one that we have already encountered. We discussed the SIS model in the context of a single population, but it also arises when describing diseases that are transmitted between two distinct populations, with an example being venereal disease transmission in a purely heterosexual population of males and females.

Here, we assume that the male population size is N and is split into two subpopulations: the susceptible male population whose size is S, and the infective male population whose size is I, such that $N = S + I$. Likewise, for the female population, we denote its total size as N^*, the female susceptible population size as S^* and the female infective population size as I^*. We will ignore changes in the population size due to birth and death, and only consider exchanges between populations due to recovery and infection. In particular, we assume the infective male population decreases at a rate aI due to recovery, while the infective female decreases at a rate a^*I^* due to recovery. Additionally, we have that the male infective population increases at a rate rSI^* due to infection, while the female infective population increases at rate r^*S^*I. The parameters a, a^*, r, r^*, are all positive constants. Putting this together, we have

$$\frac{dS}{dt} = -rSI^* + aI, \tag{15}$$

$$\frac{dS^*}{dt} = -r^*S^*I + a^*I^*, \tag{16}$$

$$\frac{dI}{dt} = rSI^* - aI, \tag{17}$$

$$\frac{dI^*}{dt} = r^*S^*I - a^*I^*. \tag{18}$$

As in the one-dimensional case, the equations for the susceptible populations are extraneous since $S = N - I$, and $S^* = N^* - I$, and accordingly, we need to only consider

$$\frac{dI}{dt} = rI^*(N - I) - aI, \tag{19}$$

$$\frac{dI^*}{dt} = r^*I(N^* - I^*) - a^*I^*. \tag{20}$$

Nondimensionalisation To begin, we nondimensionalise our equations such that

$$I = Nu, I^* = N^*v, t = \tau/a, \tag{21}$$

where u is the nondimensional male infective population, v is the nondimensional female infective population, and τ is the nondimensional time. Substituting these expressions into the differential equations gives

$$\frac{du}{d\tau} = Rv(1 - u) - u = f_1(u, v), \tag{22}$$

$$\frac{dv}{d\tau} = R^*u(1 - v) - A^*v = f_2(u, v), \tag{23}$$

where $R = rN^*/a$, $R^* = r^*N/a$, and $A^* = a^*/a$.

Nullclines Setting the right hand sides of our equations equal to zero, we find that the nullclines are given by the equations

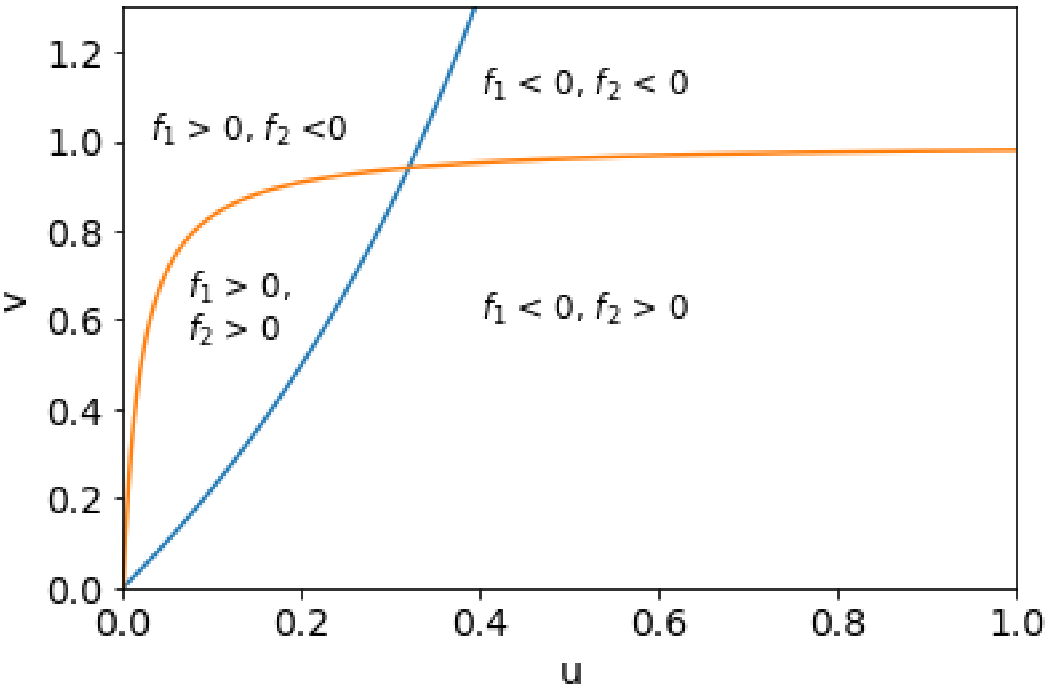
$$v = \frac{u}{R(1 - u)} \tag{24}$$

$$u = \frac{A^*v}{R^*(1 - v)}. \tag{25}$$

The code below plots the nullclines for specific parameter values. Importantly, we see that $f_1 < 0$ below the blue curve, while $f_2 < 0$ above the orange curve.

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

R = 0.5
Rs = 0.5
As = 0.01
u = np.linspace(0,0.99,1000)
v = np.linspace(0,0.99,1000)
unull = As*v/(Rs*(1 - v));
vnull = u/(R*(1 - u));

plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("u")
plt.ylabel("v")
plt.plot(u,vnull)
plt.plot(unull,v)
plt.ylim(0,1.3)
plt.xlim(0,1);
plt.text(0.03, 1, "$f_1$ > 0, $f_2$ <0", fontsize=12);
plt.text(0.4, 1.1, "$f_1$ < 0, $f_2$ < 0", fontsize=12);
plt.text(0.4, 0.6, "$f_1$ < 0, $f_2$ > 0", fontsize=12);
plt.text(0.07, 0.65, "$f_1$ > 0,", fontsize=12);
plt.text(0.07, 0.55, "$f_2$ > 0", fontsize=12);
```



Based on the intersection of the nullclines, we can observe the presence to two fixed points. The signs of f_1 and f_2 in the regions that the nullclines carve out indicate that we should expect the fixed point at the origin to be unstable, while the other fixed point should be stable.

Fixed points and their stability To find the fixed points, we seek the values of u and v that simultaneously satisfy the equations

$$Rv(1-u) - u = 0 \tag{26}$$
$$R^*u(1-v) - A^*v = 0 \tag{27}$$

We solve the first equation for v to obtain $v = u/(R(1-u))$. We then substitute this expression for v into the second equation which, after rearranging, gives the quadratic equation

$$u(R^*R - A^* - R^*(R+1)u) = 0. \tag{28}$$

This equation is satified for $u = 0$ and

$$u^* = \frac{R - A^*/R^*}{R+1}. \tag{29}$$

Using these values of u in the expression for v, we obtain $v = 0$ for $u = 0$, and

$$v^* = \frac{R^* - A^*/R}{R^* + A^*}. \tag{30}$$

when $u = (R - A^*/R^*)/(R+1)$. As u and v must be positive, we see that the nontrivial fixed point exists only when $R^*R - A^* > 0$. This is what is known as a *threshold condition* that the system's parameters must satisfy in order for a nontrivial steady state to exist. This can be confirmed by using the code above and observing that the nullclines intersect only when the threshold condition is satisfied.

While the nullclines already give some indication that $(0,0)$ will be unstable and the nontrivial fixed point will be stable, we can confirm these suspicions by performing the linear stability analysis. The Jacobian for the system is

$$J(u,v) = \begin{bmatrix} \partial f_1/\partial u & \partial f_1/\partial v \\ \partial f_2/\partial u & \partial f_2/\partial v \end{bmatrix} = \begin{bmatrix} -Rv-1 & R-Ru \\ R^*-R^*v & -R^*u-A^* \end{bmatrix} \tag{31}$$

Evaluating this at $(0,0)$, we find that

$$J(0,0) = \begin{bmatrix} -1 & R \\ R^* & -A^* \end{bmatrix}, \tag{32}$$

which has characteristic equation,

$$\lambda^2 + (1+A^*)\lambda + A^* - R^*R = 0. \tag{33}$$

The roots are given by

$$\lambda_\pm = \frac{1}{2}\left(-(1+A^*) \pm \sqrt{(1+A^*)^2 - 4(A^*-R^*R)}\right). \tag{34}$$

Thus, we see that when the threshold condition is satisfied, i.e. $R^*R - A^* > 0$, the eigenvalues are real and $\lambda_+ > 0 > \lambda_-$. Thus, $(0,0)$ is a saddle and therefore unstable.

For the nontrivial fixed point, (u^*, v^*), the analysis is also nontrivial! We can write the Jacobian as

$$J(u^*,v^*) = \begin{bmatrix} -Rv^*-1 & R-Ru* \\ R^*-R^*v* & -R^*u*-A^* \end{bmatrix} \tag{35}$$

and obtain the characteristic equation

$$\lambda^2 + (1+A^*+Rv^*+R^*u^*)\lambda + (Rv^*+1)(R^*u^*+A^*) - (R-Ru^*)(R^*-R^*v^*) = 0. \tag{36}$$

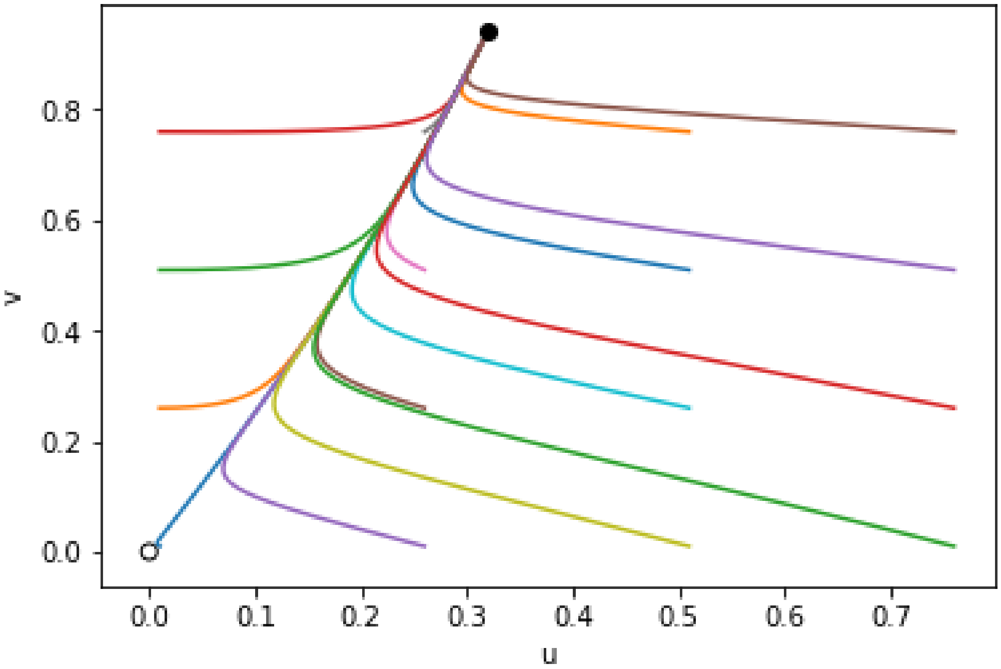
At this point, it's perhaps best to use software such as Mathematica to solve for λ. In doing so, one finds that $Re(\lambda) < 0$ for both eigenvalues, revealing that (u^*, v^*) is stable.

A good way to confirm our analysis is to compile the phase portrait numerically, which can be done using the code below.

[2]:

```python
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint

R,Rs,As = 0.5,0.5,0.01

def du_dt(u, t):
    return [R*u[1]*(1 - u[0]) - u[0], Rs*u[0]*(1 - u[1]) - As*u[1]]
```

```python
ts = np.linspace(0, 50, 1000)
ics = np.arange(0.01, 0.99, 0.25)
for i in ics:
    for j in ics:
        u0 = [i, j]
        us = odeint(du_dt, u0, ts)
        male = us[:,0]
        female = us[:,1]
        plt.plot(male,female);

plt.xlabel("u");
plt.ylabel("v");

ufp = (R - As/Rs)/(R+1)
vfp = (Rs - As/R)/(Rs + As)

plt.plot(ufp,vfp,'ko');
plt.scatter(0,0, edgecolor="black", color="none");

#    plt.plot(ts, male)
#plt.plot(ts, female);
```



We observe that the trajectories rapidly approach and then move along the unstable manifold that connects the saddle at $(0,0)$ and the stable fixed point, (u^*, v^*).

Interpretation From our analysis, we observed that the existence of the nontrivial stable fixed point hinged on the threshold condition, $R^*R - A^* > 0$, being satisfied. In the context of an epidemic, this is something to be avoided – a nontrivial stable infective population is something that one does not want! In terms of the dimensional parameters, the threshold condition can be expressed as $(rN/a)(r^*N^*/a^*) - 1 > 0$. Assuming that the number of susceptible males is N, then rN/a provides the maximum number of males that are infected for each infective female. This is found by also assuming the steady-state condition $dI/dt = 0$. We have a similar interpretation for r^*N^*/a^*: it is the maximum number of females infected for each infected male. Thus, these are the maximal contact rates for each of the populations and an epidemic can be avoided if their product is less than one.

Mathematical Biology - Week 4

October 24, 2022

0.0.1 Genetic control system

A basic model for a genetic control system consists of two chemical species: the messenger RNA (mRNA) responsible for carrying a protein's blueprint from the DNA to the cell's protein factories, and the protein encoded by the mRNA that is the intended product of the transcription process. In the control system, the generation of mRNA is stimulated by the presence of the protein via a sigmoidal function, resulting in a positive feedback loop. We will assume that the protein concentration, p, and mRNA concentration, m, degrade with rates $k_1 p$ and $k_2 m$, respectively, while the presence of mRNA leads to the creation of protein with rate $k_3 m$. The resulting equations for the protein and mRNA concentrations are

$$\frac{dp}{dt} = k_3 m - k_1 p \tag{1}$$

$$\frac{dm}{dt} = h_1 \frac{p^2}{H^2 + p^2} - k_2 m \tag{2}$$

Nondimensionalisation We first nondimensionalise our system by writing

$$p = Hu, m = h_1 v/k_2, t = \tau/k_2 \tag{3}$$

where u, v, and τ are the nondimensional protein concentration, mRNA concentration, and time, respectively. Substituting these expressions into the differential equations above yields the following system

$$\frac{du}{d\tau} = av - bu, \tag{4}$$

$$\frac{dv}{d\tau} = \frac{u^2}{1 + u^2} - v, \tag{5}$$

where $a = k_3 h_1/(H k_2^2)$ and $b = k_1/k_2$.

Nullclines The nullclines are given by the curves

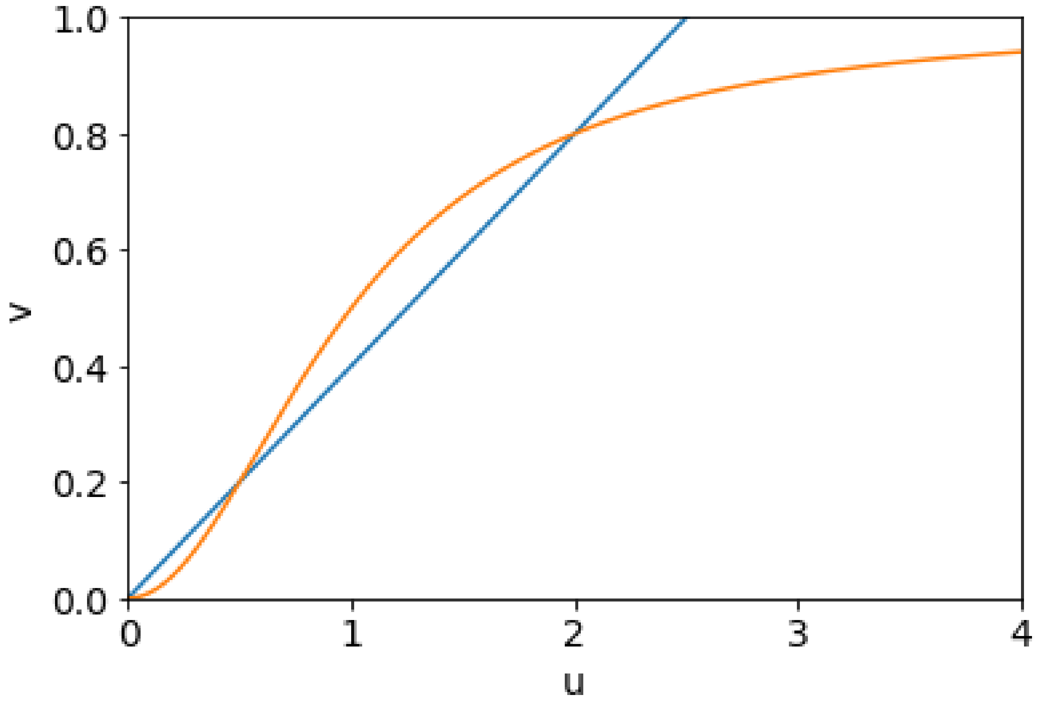
$$v = (b/a)u, \tag{6}$$

$$v = \frac{u^2}{1 + u^2}. \tag{7}$$

The nullclines can be plotted using the code below. We see that they intersect provided $\alpha = b/a < 1/2$. Can you identify the regions where f_1, f_2 are positive or negative and give some indication of the flow?

[2]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

alpha = 0.4
u = np.linspace(0,4,1000)
v1 = alpha*u
v2 = u**2/(1+u**2)

plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("u")
plt.ylabel("v")
plt.plot(u,v1)
plt.plot(u,v2)
plt.ylim(0,1)
plt.xlim(0,4);
```



Fixed points and their stability We can determine the fixed points by finding the values of u and v where the nullclines intersect. As $v = \alpha u$, this can be achieved by considering

$$\alpha u = \frac{u^2}{1+u^2}. \tag{8}$$

We see that this equation is satisfied for $u = 0$. To find the other intersections, we examine

$$\alpha u^2 - u + \alpha = 0, \tag{9}$$

which as roots

$$u_\pm = \frac{1 \pm \sqrt{1 - 4\alpha^2}}{2\alpha} \tag{10}$$

As we noted above, there are only real roots, i.e. intersections of the nullclines, if $\alpha < 1/2$. From the fact that $v = \alpha u$ at the fixed points, we see that the fixed points are $(0,0)$ and $(u_\pm, v_\pm)$, where

$$v_\pm = \frac{1 \pm \sqrt{1 - 4\alpha^2}}{2}. \tag{11}$$

To evaluate the stability of the fixed points, we compute the Jacobian,

$$J(u,v) = \left[\begin{array}{cc} -b & a \\ \frac{2u}{(1+u^2)^2} & -1 \end{array} \right]. \tag{12}$$

Evaluting the Jacobian at $(0,0)$, we obtain

$$J(0,0) = \left[\begin{array}{cc} -b & a \\ 0 & -1 \end{array} \right], \tag{13}$$

whose characteristic equation is

$$(b + \lambda)(1 + \lambda) = 0. \tag{14}$$

Thus we see that $\lambda = -1$ and $\lambda = -b$ and consequently $(0,0)$ is a stable node.

For the other fixed points, we first observe that for these fixed points, $\alpha(1 + u_\pm^2) = u_\pm$, and as such, $2u_\pm/(1 + u_\pm^2)^2 = 2\alpha^2/u_\pm$. The Jacobian is then,

$$J(u_\pm, v_\pm) = \left[\begin{array}{cc} -b & a \\ 2\alpha^2/u_\pm & -1 \end{array} \right], \tag{15}$$

whose characteristic equation is,

$$\lambda^2 + (b+1)\lambda + b\left(1 - \frac{2\alpha}{u_\pm}\right). \tag{16}$$

Substituting in the expression for $u_\pm$, the characteristic equation becomes

$$\lambda^2 + (b+1)\lambda \pm b\sqrt{1 - 4\alpha^2}. \tag{17}$$

Defining $\beta = \sqrt{1 - 4\alpha^2}$ and combined with the fact that $0 < \alpha < 1/2$, we have that $0 < \beta < 1$. Let's first consider the fixed point (u_-, v_-), which gives eigenvalues

$$\lambda_\pm = \frac{-(b+1) \pm \sqrt{(b+1)^2 + 4b\beta}}{2}. \tag{18}$$

As $\sqrt{(b+1)^2 + 4b\beta} > (b+1)$, we have that both eigenvalues are real with $\lambda_+ > 0 > \lambda_-$. Thus, (u_-, v_-) is a saddle, and therefore unstable.

For the fixed point (u_+, v_+), we have

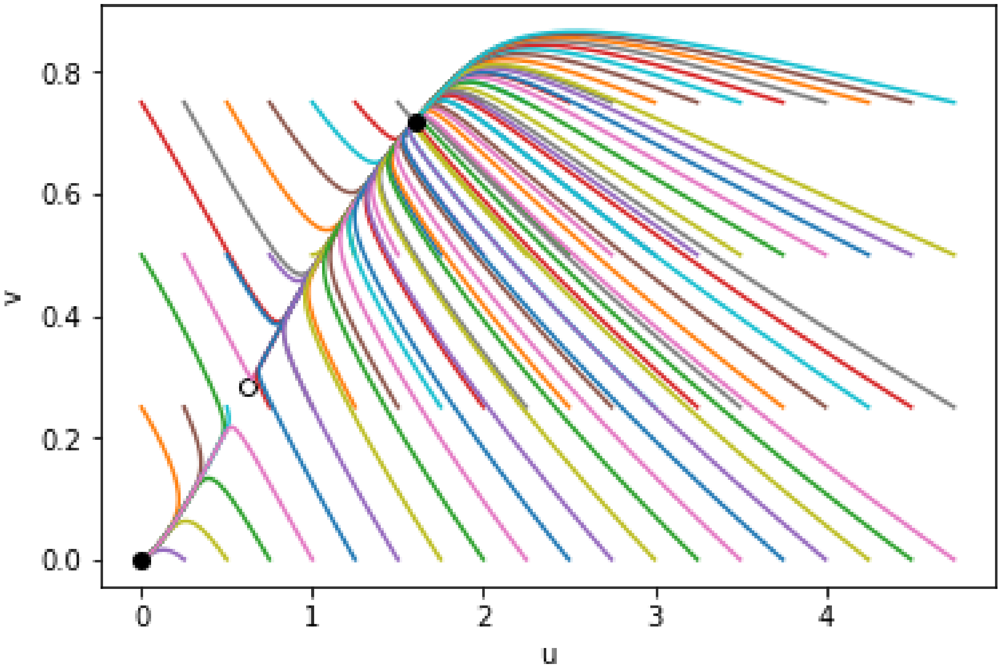
$$\lambda_\pm = \frac{-(b+1) \pm \sqrt{(b+1)^2 - 4b\beta}}{2}. \tag{19}$$

As $(b+1)^2 - 4b\beta > (b-1)^2 > 0$, we have that the eigenvalues are real. Additionally, since $\sqrt{(b+1)^2 - 4b\beta} < (b+1)$, we have $\lambda_- < \lambda_+ < 0$. Thus, the fixed point (u_+, u_-) is a stable node.

We see that for $\alpha < 1/2$, we have a bistability with $(0,0)$ and (u_+, v_+) being stable, while the 'middle' fixed point (u_-, v_-) is unstable. The phase portrait can be generated numerically by running the following code:

[3]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
import math

alpha = 0.45
b = 1
a = b/alpha

def du_dt(u, t):
    return [a*u[1] - b*u[0], u[0]**2/(1 + u[0]**2) - u[1]]

ts = np.linspace(0, 50, 1000)
icsu = np.arange(0.0, 5, 0.25)
icsv = np.arange(0.0, 1, 0.25)
for i in icsu:
    for j in icsv:
        u0 = [i, j]
        us = odeint(du_dt, u0, ts)
        protein = us[:,0]
        mRNA = us[:,1]
        plt.plot(protein,mRNA);

plt.plot(0,0,'ko');

if alpha < 0.5:
    up = (1 + math.sqrt(1 - 4*alpha**2))/(2*alpha)
    vp = alpha*up

    um = (1 - math.sqrt(1 - 4*alpha**2))/(2*alpha)
    vm = alpha*um
    plt.plot(up,vp,'ko');
    plt.scatter(um,vm, edgecolor="black", color="none");
plt.xlabel("u");
plt.ylabel("v");
```



Interpretation We see that bistability indicates that depending on the intial concentrations of the protein and mRNA, protein production will go to zero, or the concentration of protein will be sustained at a level set by the system. This provides a switching mechanism for the gene, which will turn on to sustain a sufficiently high concentration of the protein, or will remain off if the ambient protein concentration is too low.

We see something else that is interesting – the situation is very similar to what we've observed for bistability in one dimensional systems with the x-axis replaced by the unstable manifold that connects the unstable fixed point with the two stable ones. We see that trajectories rapidly approach the unstable manifold and flow nearly along it toward one of the stable fixed points. Additionally, we'll see soon that the bifurcation that gives rise the fixed points $(u_\pm, v_\pm)$ is very similar to what we have already encountered in one dimensional systems.

0.0.2 Competition between populations

Often in nature, multiple species compete for the same resources and as a result, the presence of one population impedes the growth of another. Let's take the case of 2 species whose population sizes are N_1 and N_2, respectively. In isolation, each population will evolve according to logistic growth. When the other population is present, the death rate of a species will be proportional to the population size of the other species. Accordingly, we have the following system

$$\frac{dN_1}{dt} = r_1 N_1 \left(1 - \frac{N_1}{K_1} - b_{12}\frac{N_2}{K_1}\right) \tag{20}$$

$$\frac{dN_2}{dt} = r_2 N_2 \left(1 - \frac{N_2}{K_2} - b_{21}\frac{N_1}{K_2}\right) \tag{21}$$

Nondimensionalisation We begin by nondimensionalising. We nondimensionalise the system by expressing the population sizes and time as

$$N_1 = K_1 u_1, N_2 = K_2 u_2, \text{ and } t = \frac{1}{r_1}\tau, \tag{22}$$

where u_1 and u_2 are the nondimensional population sizes, and τ is the nondimensional time. Substituting this into the original system, we have

$$\frac{du_1}{d\tau} = u_1\left(1 - u_1 - a_{12}u_2\right), \tag{23}$$

$$\frac{du_2}{d\tau} = \rho u_2\left(1 - u_2 - a_{21}u_1\right), \tag{24}$$

where $\rho = r_2/r_1$, $a_{12} = b_{12}K_2/K_1$, and $a_{21} = b_{21}K_1/K_2$.

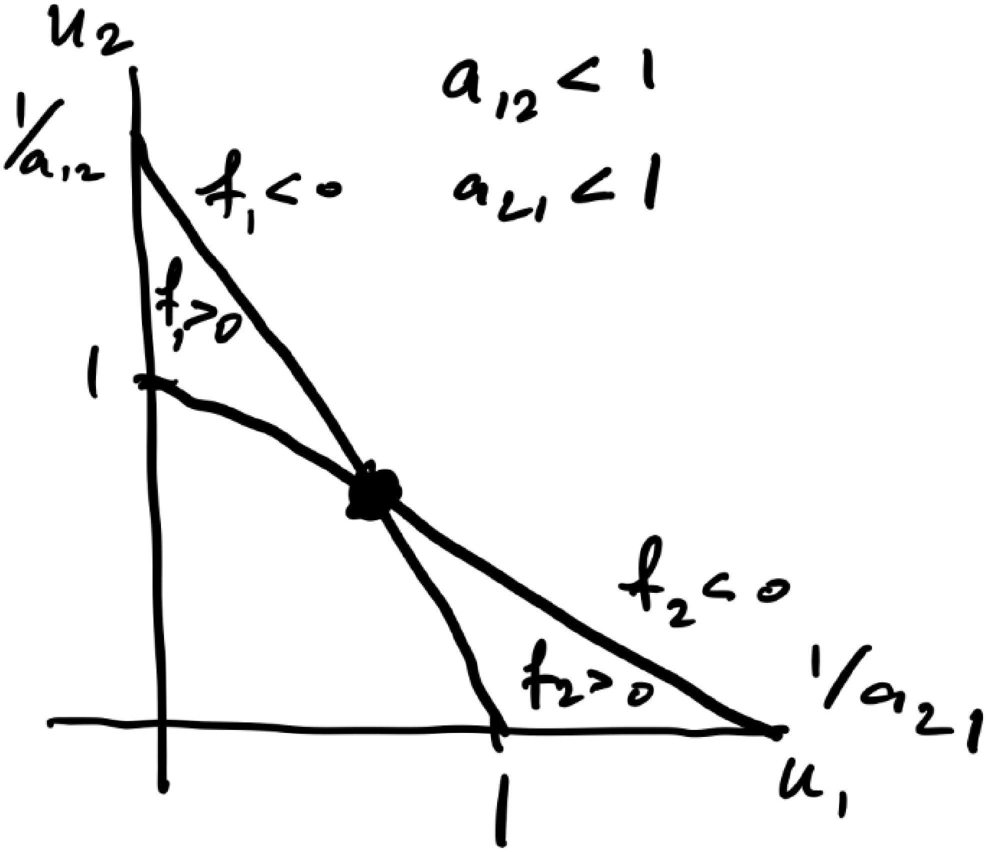
Nullclines To find the nullclines, we consider

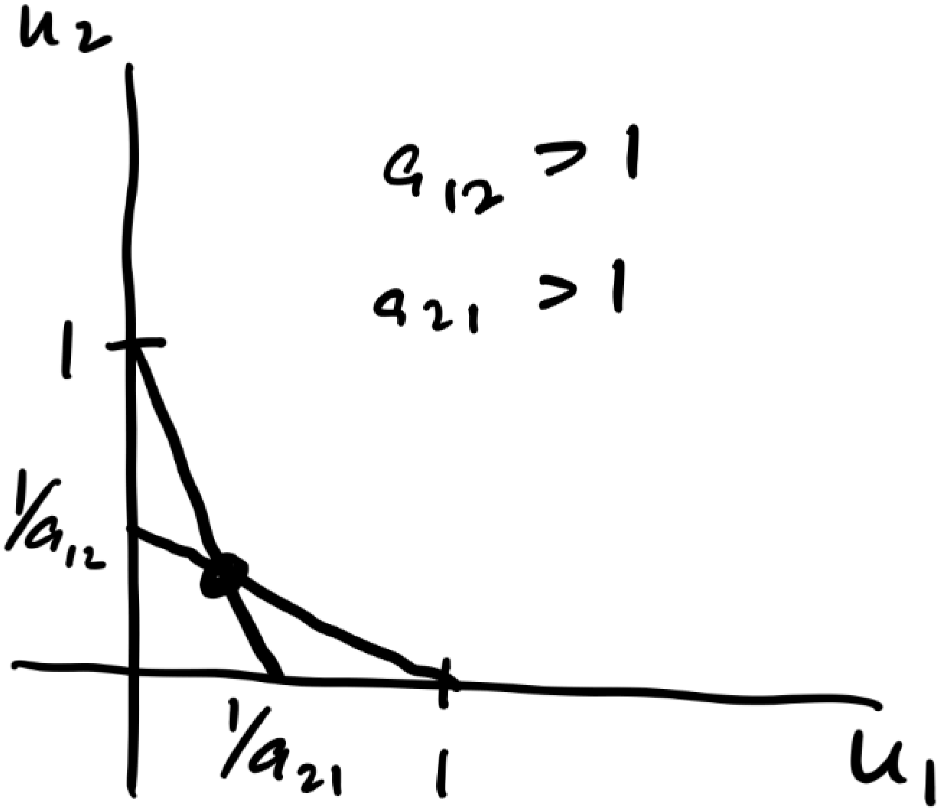
$$u_1(1 - u_1 - a_{12}u_2) = 0 \tag{25}$$

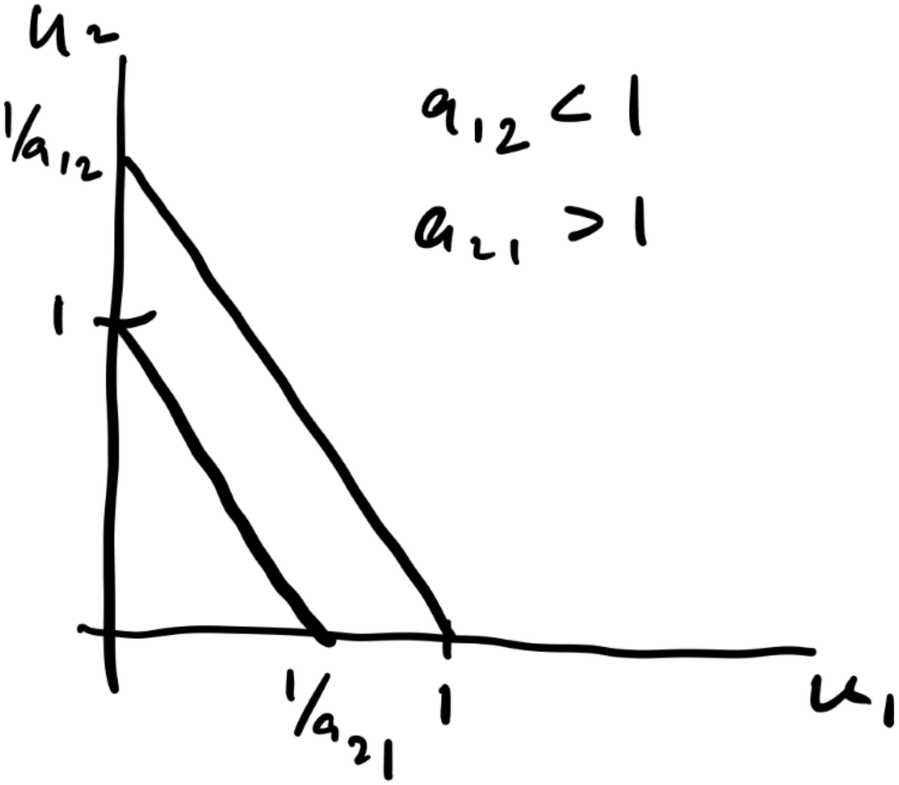
and

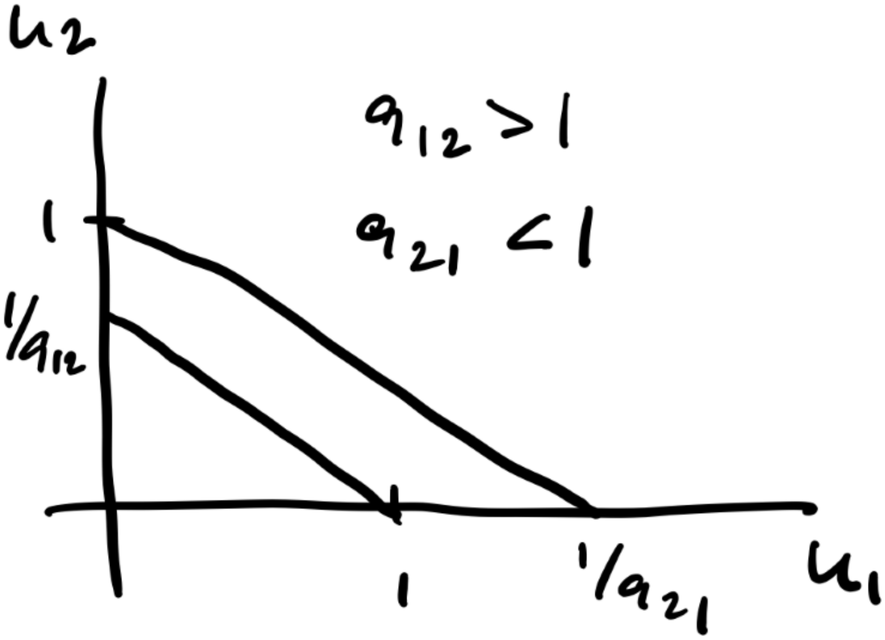
$$\rho u_2(1 - u_2 - a_{21}u_1) = 0, \tag{26}$$

separately. In the first case, we see that the equation is satisfied along the lines $u_1 = 0$ and $u_1 = 1 - a_{12}u_2$. The second equation is satisfied when $u_2 = 0$, or $u_2 = 1 - a_{21}u_1$. We see that the slopes of the lines, and whether they intersect or not, depend on the values of a_{21} and a_{12}. In particular, we see that four cases can arise and these are shown in the sketches below.


u_2
$1/a_{12}$
$a_{12} < 1$
$f_1 < 0$
$a_{21} < 1$
$f_1 > 0$
1
$f_2 < 0$
$f_2 > 0$
$1/a_{21}$
u_1
1


u_2
$a_{12} > 1$
$a_{21} > 1$
1
$1/a_{12}$
$1/a_{21}$
1
u_1





We see that there are two cases where these two nullclines intersect, and two where they do not. As the intersections are the fixed points, we see that varying a_{21} and a_{12} result in bifurcations.

Fixed points and their stability By considering

$$u_1(1 - u_1 - a_{12}u_2) = 0 \tag{27}$$
$$\rho u_2(1 - u_2 - a_{21}u_1) = 0, \tag{28}$$

simultaneously, we find that the fixed points are

$$u_1^* = 0, u_2^* = 0; u_1^* = 1, u_2^* = 0; u_1^* = 0, u_2^* = 1; \tag{29}$$
$$u_1^* = \frac{1 - a_{12}}{1 - a_{12}a_{21}}, u_2^* = \frac{1 - a_{21}}{1 - a_{12}a_{21}}. \tag{30}$$

We can then go ahead an compute the Jacobian to find,

$$J(u_1, u_2) = \begin{bmatrix} 1 - 2u_1 - a_{12}u_2 & -a_{12}u_1 \\ -\rho a_{21}u_2 & \rho(1 - 2u_2 - a_{21}u_1) \end{bmatrix}. \tag{31}$$

Evaluating this at $(0, 0)$, we obtain

$$J(0, 0) = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix}, \tag{32}$$

and thus, $(0, 0)$ is an unstable node regardless of the parameter values. This means that mass extinction is impossible – there will always be a population, though the exact composition remains unclear as of yet.

Let's now take the fixed point $(1, 0)$, for which the Jacobian is

$$J(1, 0) = \begin{bmatrix} -1 & -a_{12} \\ 0 & \rho(1 - a_{21}) \end{bmatrix}, \tag{33}$$

which as eigenvalues $\lambda_1 = -1$ and $\lambda_2 = \rho(1 - a_{21})$. Thus, the stability hinges on the magnitude of a_{21}. If $a_{21} > 1$, then $(1, 0)$ is a stable node, while if $a_{21} < 1$, $(1, 0)$ is a saddle point and hence unstable.

Following very similar arguments for $(0, 1)$, we have $\lambda_1 = -\rho$, and $\lambda_2 = 1 - a_{12}$ and consequently, $(0, 1)$ is a stable node if $a_{12} > 1$ and saddle point if $a_{12} < 1$.

Finally, for $u_1^* = (1 - a_{12})/(1 - a_{12}a_{21})$, and $u_2^* = (1 - a_{21})/(1 - a_{12}a_{21})$ the Jacobian is

$$J = (1 - a_{21}a_{12})^{-1} \begin{bmatrix} a_{12} - 1 & a_{12}(a_{12} - 1) \\ \rho a_{21}(a_{21} - 1) & \rho(a_{21} - 1) \end{bmatrix}. \tag{34}$$

The eigenvalues are given by the lengthy expression

$$\lambda_{1,2} = \frac{a_{12} - 1 + \rho(a_{21} - 1) \pm \sqrt{(a_{12} - 1 + \rho(a_{21} - 1))^2 - 4\rho(1 - a_{12}a_{21})(a_{12} - 1)(a_{21} - 1)}}{2(1 - a_{12}a_{21})}. \tag{35}$$

Unfortunately, this expression is a bit challenging to examine at first glance, but after a bit of thought, the stability can be revealed. First, we know that the fixed point exists only if $a_{12} > 1$ and $a_{21} > 1$, or $a_{12} < 1$ and $a_{21} < 1$. Let's take $a_{12} < 1$ and $a_{21} < 1$. As a result, $a_{12} - 1 + \rho(a_{21} - 1) < 0$ and $1 - a_{12}a_{21} > 0$. Additionally, we have that $0 < (a_{12} - 1 - \rho(a_{21} - 1))^2 < (a_{12} - 1 + \rho(a_{21} - 1))^2 - 4\rho(1 - a_{12}a_{21})(a_{12} - 1)(a_{21} - 1) < (a_{12} - 1 + \rho(a_{21} - 1))^2$. Therefore, we know that both eigenvalues will be real and $\lambda_{1,2} < 0$. The fixed point is then a stable-node for $a_{12} < 1$ and $a_{21} < 1$.

Now, if we consider $a_{12} > 1$ and $a_{21} > 1$, we have that $a_{12} - 1 + \rho(a_{21} - 1) > 0$, however $1 - a_{12}a_{21} < 0$. We also see that $0 < (a_{12} - 1 + \rho(a_{21} - 1))^2 < (a_{12} - 1 + \rho(a_{21} - 1))^2 - 4\rho(1 - a_{12}a_{21})(a_{12} - 1)(a_{21} - 1)$. Accordingly, we see then that $\lambda_1 < 0 < \lambda_2$, revealing that the fixed point is a saddle if $a_{12} > 1$ and $a_{21} > 1$.

Phase portrait With the fixed points and their stability established, we can combine this with what we know already about the nullclines to assemble a picture of the dynamics for the four different cases, as well as describe their ecological implications.

Before doing so, it's worth noting that along the lines $u_1 = 0$, $u_1 = U_1$, $u_2 = 0$ and $u_2 = U_2$, where constants $U_1, U_2 > 1$ the vector (f_1, f_2) points inward or along the boundary. Hence the system is globally stable.

Now let's look more closely at the four cases, each of which can be considered numerically using the code in the cell below.

1. $a_{12} < 1, a_{21} < 1$: Here, both $(1, 0)$ and $(0, 1)$ are unstable, while the fourth fixed point, (u_1^*, u_2^*), will be a stable-node. Thus, we see that for this case, where mutual competition is sufficiently low, both species will coexist regardless of the initial populations.

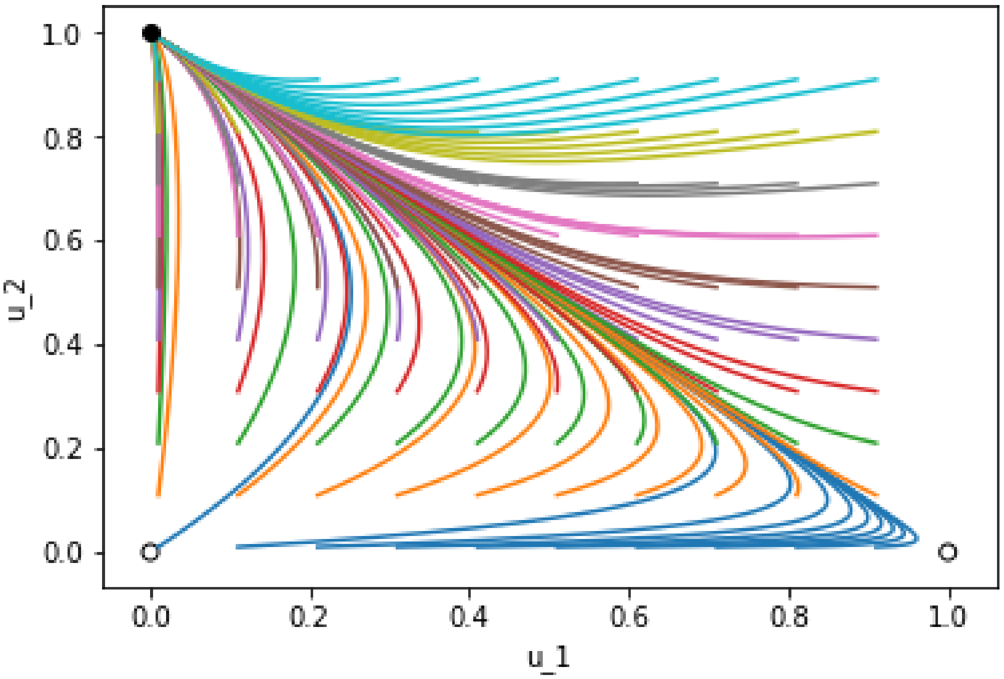
2. $a_{12} > 1, a_{21} > 1$: For this case, both $(1, 0)$ and $(0, 1)$ are stable, while the fourth fixed point, (u_1^*, u_2^*), turns out to be a saddle point as $1 - a_{12}a_{21} < 0$. Thus, trajectories will move towards either $(1, 0)$ or $(0, 1)$ depending on the initial condition. The stable manifold for the saddle forms the separatrix, which is the boundary of the domains of attraction for $(1, 0)$ and $(0, 1)$. Thus, where mutual competition is strong, only one population survives, while the other dies out, and which wins depends on the initial population sizes.

3. $a_{12} < 1, a_{21} > 1$: In this case, we have $(1, 0)$ is stable, while $(0, 1)$ is unstable. The fourth steady state does not exist, indicated by the fact that the nullclines do not intersect. We see then that as a result of a disproportionate competition, where the presence of species 1 negatively affects species 2 so much so that species 2 becomes extinct regardless of the initial condition, and species 1 reaches its carrying capacity.

4. $a_{12} > 1, a_{21} < 1$: This case is the same as case 3, but with species 1 exchanged for species 2.

[3]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint

rho,a12,a21 = 1,1.5,0.5

def du_dt(u, t):
    return [u[0]*(1 - u[0] - a12*u[1]), rho*u[1]*(1 - u[1] - a21*u[0])]

ts = np.linspace(0, 50, 1000)
ics = np.arange(0.01, 0.99, 0.1)
for i in ics:
    for j in ics:
        u0 = [i, j]
        us = odeint(du_dt, u0, ts)
        pop1 = us[:,0]
        pop2 = us[:,1]
        plt.plot(pop1,pop2);
plt.xlabel("u_1");
plt.ylabel("u_2");

plt.scatter(0,0, edgecolor="black", color="none");

u1s = (1 - a12)/(1 - a12*a21);
u2s = (1 - a21)/(1 - a12*a21);

if a12 < 1:
    if a21 < 1:
        plt.scatter(1,0, edgecolor="black", color="none");
        plt.scatter(0,1, edgecolor="black", color="none");
        plt.plot(u1s,u2s,'ko');
    else:
        plt.plot(1,0,'ko');
        plt.scatter(0,1, edgecolor="black", color="none");
else:
    if a21 > 1:
        plt.scatter(u1s,u2s, edgecolor="black", color="none");
        plt.plot(1,0,'ko');
        plt.plot(0,1,'ko');
    else:
        plt.plot(0,1,'ko');
        plt.scatter(1,0, edgecolor="black", color="none");
```



Interpretation A rather striking feature of this model is that 3 of the 4 cases described above result in exinction of one of the species! The exinction of one population due to competition with another is known as the *principle of competitive exclusion.* In our model, we see that whether this occurs or not depends on the groups $a_{12} = b_{12}K_2/K_1$ and $a_{21} = b_{21}K_1/K_2$, and hence the competition coefficients and carrying capacities. Consider a population of large animals, N_1, and a population of smaller animals, N_2, that compete for the same food source, such as grass in a fixed area. Suppose that $b_{21} = b_{12}$. As the carrying capacity of the land is lower for the larger animals, we have that $K_1 < K_2$, and as a result $a_{12} > a_{21}$. We can imagine then that we may encounter $a_{12} > 1$ and $a_{21} < 1$, which will result in the extinction of the larger animals leaving the smaller animal population to flourish.

0.0.3 Predator-Prey systems

In our final example, we will examine multi-species population dynamics through the classic **Lotka-Volterra** system for a predator-prey system that was originally developed by Volterra in 1926 to explain the oscillatory levels of certain fish catches in the Adriatic. Let the population size of prey be given by $N(t)$ and $P(t)$ be the size of the predator population. The Lotka-Volterra system is

$$\frac{dN}{dt} = aN - bNP, \tag{36}$$

$$\frac{dP}{dt} = -dP + cNP. \tag{37}$$

where a, b, c, and d are positive constants. We see that in the absence of interactions ($b = c = 0$), the prey population will grow exponentially with birth rate aN, while the predator population size will shrink exponentially with death rate $-dP$. The interactions between the populations will allow the predator population to grow at rate cNP, while the prey population decreases with rate $-bNP$.

The reason that this is called the Lotka-Volterra system rather than just the Volterra system is that Lotka derived the same set of equations to explain oscillations appearing in a chemical system. Using the law of mass action, show that the reaction

$$A + X \xrightarrow{k_1} 2X, X + Y \xrightarrow{k_2} 2Y, Y \xrightarrow{k_3} B \tag{38}$$

with constant concentration of A, a, yields the Lotka-Volterra system.

Nondimensionalisation Let's go ahead an nondimensionalise our system through

$$N = \frac{d}{c}u, P = \frac{a}{b}v, \text{ and } t = \frac{1}{a}\tau \tag{39}$$

where u is the nondimensional prey population size, v is the nondimensional predator population size, and τ is nondimensional time. Substituting these expressions into (37), we obtain

$$\frac{du}{d\tau} = u(1 - v), \tag{40}$$

$$\frac{dv}{d\tau} = \alpha v(u - 1). \tag{41}$$

Fixed points and their stability Examining the right hand side reveals that there are two fixed points, $u = v = 0$, and $u = v = 1$. To characterise their stability, we compute the Jacobian,

$$J(u,v) = \begin{bmatrix} 1-v & -u \\ \alpha v & \alpha(u-1) \end{bmatrix} \tag{42}$$

and evaluating it at $u = v = 0$, we see that

$$J(0,0) = \begin{bmatrix} 1 & 0 \\ 0 & -\alpha \end{bmatrix}. \tag{43}$$

Conveniently, there is no need to diagonalise and we see immediately that since $\alpha > 0$, we have that the origin in the uv-plane is a saddle point and as a consequence, unstable. The Jacobian evaluated at $(1,1)$ gives

$$J(1,1) = \begin{bmatrix} 0 & -1 \\ \alpha & 0 \end{bmatrix}. \tag{44}$$

This one we'll need to diagonalise, but upon doing so, we see that $\lambda_{1,2} = \pm i\sqrt{\alpha}$. Thus, $(1,1)$ is a centre whose stability is marginal and typically cannot be determined from linear analysis alone (notice that the stability theorem says nothing about this case!).

Analytical solution In this case, the analytical solution for the phase plane trajectories is possible. If we consider $dv/d\tau = (dv/du)(du/d\tau)$, we obtain the separable equation

$$\frac{dv}{du} = \alpha\frac{v(u-1)}{u(1-v)}, \tag{45}$$

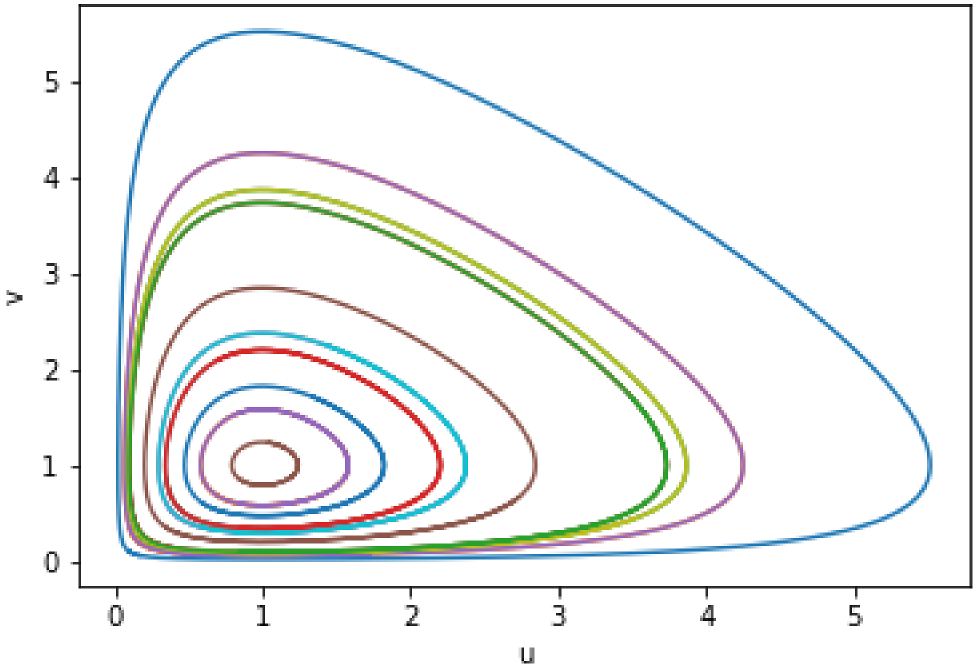
which upon integration yields the family of curves

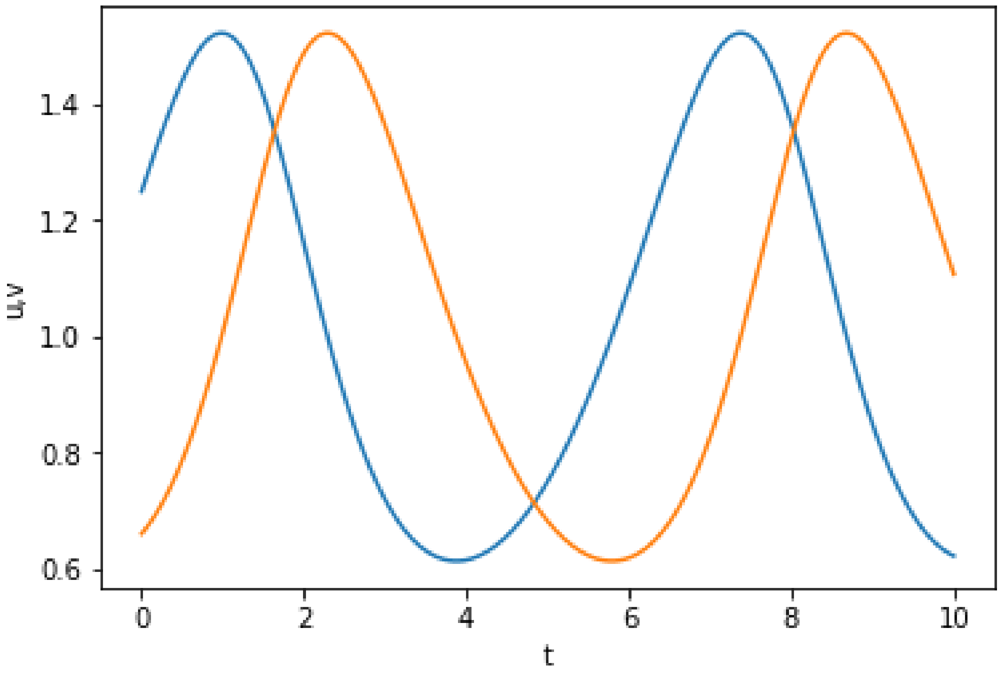
$$\alpha u + v - \log(u^{\alpha} v) = H, \tag{46}$$

where H is the constant of integration whose exact value will depend upon the initial conditions. The phase portrait is plotted below for different initial conditions. The second figure shows the populations as a function of time.

[4]:

```python
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint

alpha = 1;

def du_dt(u, t):
    return [u[0]*(1 - u[1]), alpha*u[1]*(u[0] - 1)]

ts = np.linspace(0, 10, 1000)
ics = np.arange(0.1, 0.9, 0.25)
for i in ics:
    for j in ics:
        u0 = [i, j]
        us = odeint(du_dt, u0, ts)
        pop1 = us[:,0]
        pop2 = us[:,1]
        plt.figure(0)
        plt.plot(pop1,pop2);

plt.xlabel("u");
plt.ylabel("v");

u0 = [1.25, 0.66]
us = odeint(du_dt, u0, ts)
pop1 = us[:,0]
pop2 = us[:,1]
plt.figure(1)
plt.plot(ts,pop1)
plt.plot(ts,pop2);
plt.xlabel("t");
plt.ylabel("u,v");
```





The solution shows that the oscillatory behaviour occuring close to the fixed point persists further away, and in fact, there are many closed orbits surrounding the centre. We see that Volterra was successful in producing a model that gives oscillations, however, its not without its flaws! Changing the initial condition is equivalent to varying the constant H that define each of the orbits. In terms of the biology, this implies a couple of things: 1. there is no natural oscillation in the population levels – different initial population sizes yield different oscillations and 2. this causes strange, unnatural things to occur, such as lowering the initial population of predators resulting in larger peaks in their population size.

There are also deficiencies from a mathematical perpspective. This model is what is known as *structurally unstable* a property which is linked to the neutral stability of the centre fixed point. Namely, a perturbation in the equation itself will lead to a drastic change in the nature of the solution. In this case, the slightest bit of damping will change the centres to stable spirals by creating a small, but non-zero, real eigenvalue.

We'll come to the resolution of these issues in the next section when we discuss oscillations and see that both the biological and mathematical issues can be resolved by introducing a more realistic population dynamics through logistic growth.

Mathematical Biology - Week 5

November 1, 2022

1 Oscillations and bifurcations

Oscillatory or periodic behaviour is fundamental to biological systems. You need to look no further than your own body to notice examples of periodic rhythms and patterns, such as the beating of our hearts, the inflation and deflation of our lungs, or the circadian cycle governing being asleep or awake, to name just a few. For each of these processes, we can pinpoint a particular frequency and amplitude with which the oscillations occur. This is in contrast with the Lotka-Volterra system where we observed the oscillations, in particular their amplitude, are instead highly dependent upon the initial conditions, with many closed orbits circling the centre fixed point. We'll now examine the existence of isolated closed trajectories, or limit cycles, and study how to predict their frequency and amplitude under certain assumptions.

Finally, we'll revist bifurcations in the context of multidimensional systems. While we will see that the bifurcations we encountered in one-dimensional systems exist in a very similar way in the multidimensional context, new types of bifurcations can occur. In particular, we'll examine the case of the (supercritical) Hopf bifurcation where upon the variation of a parameter, a stable fixed point becomes unstable and the new attractor is a stable limit cycle.

At each step of the way, we'll explore models of biological phenomena that exhibit limit cycles, interpreting their role in the biological process that the model is meant to capture.

1.1 Existence of closed orbits: Poincaré-Bendixson Theorem

We will begin our discussion by answering the question: how can we know if a closed orbit exists? The answer to this question for 2D systems is provided by the *Poincaré-Bendixson Theorem* which we state here.

Poincaré-Bendixson Theorem Suppose that:

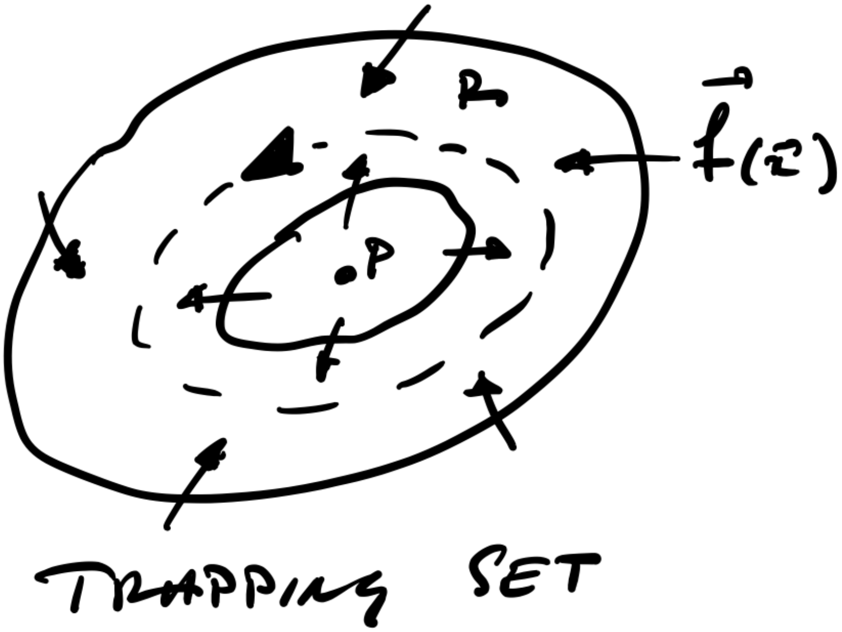
1. R is a closed, bounded subset of the plane
2. $d\mathbf{x}/dt = \mathbf{f}(\mathbf{x})$ is a continuously differentiable vector field on an open set containing R
3. R does not contain any fixed points
4. There exists a trajectory C that is confined in R, meaning that it begins in R and remains in R for all future times.

Then either C is a closed orbit, or it spirals toward a closed orbit at $t \to \infty$.

We won't prove this theorem, but we'll need to understand how to apply it to ascertain the existence of closed orbits. In doing so, we need to ensure that each of the conditions described above is met.

While the first three can be easily satisfied or checked, the fourth condition normally requires a bit more effort.

The usual approach is to ensure that R is chosen such that it is a *trapping set*, meaning that the flow points inward along the boundary of R. More specifically, if $\mathbf{n}(\mathbf{x})$ is the outward-pointing unit-vector along the boundary of R, ∂R, then R will be a trapping set if $\mathbf{f}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0$ for all $\mathbf{x} \in \partial R$. Constructing the trapping set ensures that all trajectories in R are confined, not just one. Provided that our trapping set does not contain a fixed point and the continuity conditions on $\mathbf{f}$ are satisfied, the Poincaré-Bendixson Theorem will guarantee that R contains a closed orbit.



1.1.1 More-realistic predator-prey system

Previously, we observed several biologically unrealistic and mathematically undesirable properties of the Lotka-Volterra system. Both of these deficiencies can be alleviated by using a more realistic predator-prey model that incorporates logistic growth and a nonlinear predation term,

$$\frac{dN}{dt} = N\left(r\left(1 - \frac{N}{K}\right) - k\frac{P}{N+D}\right), \tag{1}$$

$$\frac{dP}{dt} = sP\left(1 - h\frac{P}{N}\right), \tag{2}$$

where r, K, k, D, s and h are positive constants. We see that in the absence of predator population, $P(t)$, the prey populations size, $N(t)$, obeys logistic growth. The predation term is similar to that used in the spruce budworm model in that for low N, predation will increase as the prey population size increases, but for large N predation is largely independent of N. The predator population is taken to follow a logistic-like equation, however, the carrying capacity is proportional to the size of the prey population, rather than simply being a constant.

In this example, we'll use the Poincaré-Bendixson Theorem to determine the conditions for which this model has a limit cycle.

Nondimensionalisation We can nondimensionalise the system by introducing

$$N = Ku, P = Kv/h, t = \tau/r \tag{3}$$

such that the system becomes

$$\frac{du}{d\tau} = u(1-u) - \frac{auv}{u+d} = f_1(u,v) \tag{4}$$
$$\frac{dv}{d\tau} = bv\left(1 - \frac{v}{u}\right) = f_2(u,v) \tag{5}$$

where $a = k/(hr)$, $b = s/r$, and $d = D/K$.

Nullclines and fixed points Previously, we determined the fixed points and nullclines in order to obtain a general picture of the flow in phase space. We'll now use them to aid us in constructing our trapping set.

The nullclines are given by the curves

$$0 = u(1-u) - \frac{auv}{u+d} \tag{6}$$
$$0 = bv\left(1 - \frac{v}{u}\right) \tag{7}$$

From the first equation ($f_1 = 0$), we have that

$$v = (1-u)(u+d)/a, \tag{8}$$

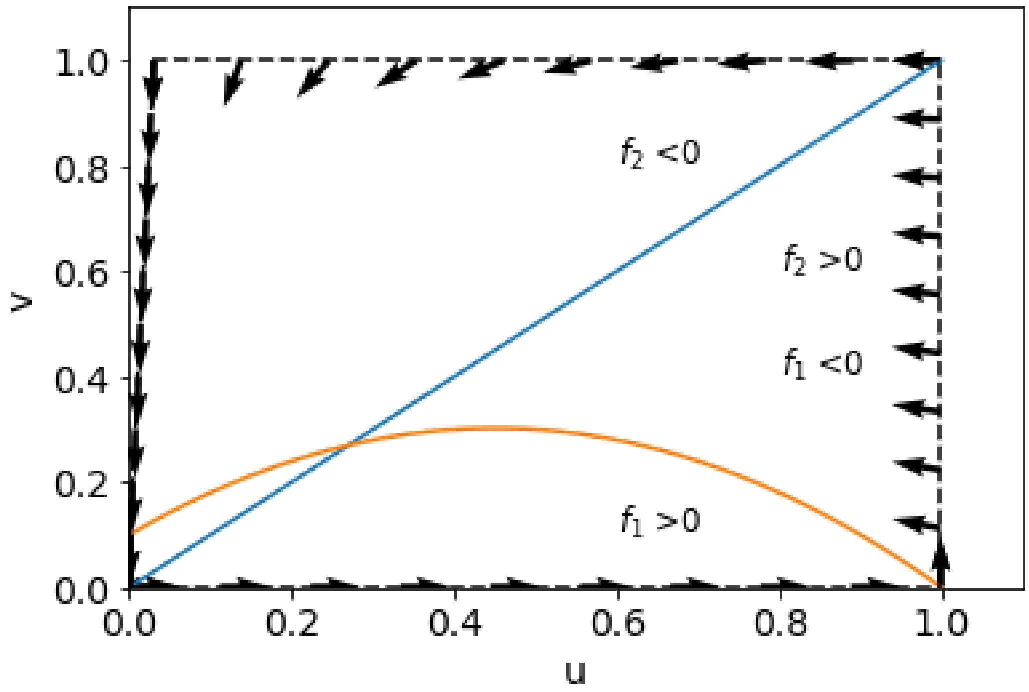
while the second ($f_2 = 0$) yields $v = u$ and $v = 0$. The nullclines can be plotted using the python cell below for different values of a, b and d.

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

#Set the parameter values
```

```
a = 1.0;
b = 0.2;
d = 0.1;

#Compute and plot the nullclines
u = np.linspace(0,1,1000)
v1 = u
v2 = (1 - u)*(u + d)/a

plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("u")
plt.ylabel("v")
plt.plot(u,v1)
plt.plot(u,v2)
plt.ylim(0,1.1)
plt.xlim(0,1.1);
plt.text(0.6, 0.8, "$f_2$ <0", fontsize=12);
plt.text(0.8, 0.6, "$f_2$ >0", fontsize=12);
plt.text(0.8, 0.4, "$f_1$ <0", fontsize=12);
plt.text(0.6, 0.1, "$f_1$ >0", fontsize=12);


#Set the corners of the trapping set.
ub = 0.999
vb = ub
ua = ub
va = (1 - ua)*(ua + d)/a
ue = va;
ve = va;
ud = ue;
vd = (1 - ud)*(ud + d)/a
vc = vb
uc = 0.03

#Set the right-most boundary of the trapping set
u = np.linspace(ua,ub,10)
v = np.linspace(va,vb,10)
f1 = u*(1-u) - a*u*v/(u+d)
f2 = b*v*(1 - v/u)
norm = (f1**2 + f2**2)**0.5
plt.plot(u,v,'k--')
plt.quiver(u,v,f1/norm,f2/norm)

#Set the lower boundary of the trapping set
u = np.linspace(ue,ua,10)
v = np.linspace(ve,va,10)
f1 = u*(1-u) - a*u*v/(u+d)
```

```
f2 = b*v*(1 - v/u)
norm = (f1**2 + f2**2)**0.5
plt.plot(u,v,'k--')
plt.quiver(u,v,f1/norm,f2/norm)

#Set part of the left boundary
u = np.linspace(ue,ud,3)
v = np.linspace(ve,vd,3)
f1 = u*(1-u) - a*u*v/(u+d)
f2 = b*v*(1 - v/u)
norm = (f1**2 + f2**2)**0.5
plt.plot(u,v,'k--')
plt.quiver(u,v,f1/norm,f2/norm)

#Set the upper boundary
u = np.linspace(uc,ub,10)
v = np.linspace(vc,vb,10)
f1 = u*(1-u) - a*u*v/(u+d)
f2 = b*v*(1 - v/u)
norm = (f1**2 + f2**2)**0.5
plt.plot(u,v,'k--')
plt.quiver(u,v,f1/norm,f2/norm)

#Find and plot the diagonal line
u = np.linspace(ud,uc,10)
m = (vc - vd)/(uc - ud)
k = vd - m*ud
v = m*u + k
f1 = u*(1-u) - a*u*v/(u+d)
f2 = b*v*(1 - v/u)
norm = (f1**2 + f2**2)**0.5
plt.plot(u,v,'k--')
plt.quiver(u,v,f1/norm,f2/norm);
```



Using the nullclines, we determine the regions where $f_1 > 0$ or $f_1 < 0$, and $f_2 > 0$ or $f_2 < 0$ and subsequently use this information to construct the outer region of the trapping set, R.

Let's take, for example, the line $u = 1$ as a possible right most boundary of the trapping set. Along this line, we know that the outward pointing normal will be $\widehat{\mathbf{u}}$, the unit vector pointing to the right. Thus, when considering $\mathbf{n} \cdot \mathbf{f} < 0$ is satisfied, for this boundary, we'll have to check if $\widehat{\mathbf{u}} \cdot \mathbf{f}(1, v) = f_1(1, v) < 0$. Since,

$$f_1(1, v) = -\frac{av}{d+1}, \tag{9}$$

$a > 0$, and $d > 0$, we have $f_1(1, v) < 0$ for $v > 0$ and the condition for the trapping set is satisfied (at least for this part of the boundary!). Notice that as long as the boundary that we are constructing is in the region $f_1 < 0$, we could have taken any line $u = k$, where k is a constant. In the Python cell above, we take $u = 0.999$

Similarly, we can take the line $v = 1$. Here, $\mathbf{n} = \widehat{\mathbf{v}}$, and we are therefore interested in determining if $f_2 < 0$ along this line. For $v = 1$, we have

$$f_2(u, 1) = b(1 - 1/u). \tag{10}$$

Since, $b > 0$ and provided that $u < 1$, we'll have $f_2 < 0$ along this line. Note, that these line cut the nullclines in such a way that flow will be along the boundary, but this is admissible since R

is closed. Again, provided we are in the region $f_2 < 0$, we can take any line $v = k$, where k is a constant.

These and the three additional sections of the outer boundary of R are plotted using the Python cell above. The arrows along the lines indicate the direction of the vector $\mathbf{f}$ at the point where it is drawn. Two of the addition sections are constructed in a very similar way using vertical or horizontal lines. The final section requires a diagonal line as $f_1 < 0$ in this region of phase space. This means a vertical line at this part of left boundary would not necessarily yield a trapping set. Note that finding this diagonal line is a nontrivial task!

So far, so good, but there is a slight problem – the set that we constructed contains a fixed point! To get rid of it we use a little trick. We 'puncture' the set that we created by introducing an infinitesimally small circle drawn around the fixed point. The next step is to guarantee that the flow is directed into R along the edge of the infinitesimal hole that we introduced.

Taking advantage of what we know about the stability of the fixed points, we can guarantee the flow will be inward provided that $Re(\lambda_i) > 0$ for $i = 1, 2$, i.e. the fixed point is an unstable node or unstable spiral (note that a saddle does not apply!).

First, let's find the precise values of u and v at the fixed point. To do this, we first set $v = u$ and consider

$$u = (1 - u)(u + d)/a \tag{11}$$

which yields

$$u^* = \frac{(1 - a - d) + \sqrt{(1 - a - d)^2 + 4d}}{2}. \tag{12}$$

and $v^* = u^*$.

To assess the stability of the fixed point, we proceed in the usual way by computing the Jacobian and evaluate it at the fixed point (u^*, u^*). This gives

$$J(u^*, u^*) = \begin{bmatrix} u^* \left(\frac{au^*}{(u^*+d)^2} - 1 \right) & -\frac{au^*}{u^*+d} \\ b & -b \end{bmatrix} \tag{13}$$

where we have used the fact that $u^*(1 - u^*) = au^*v^*/(u^* + d)$.

Now we'd like to find under which conditions are real parts of both eigenvalues of this matrix greater than zero. Rather than computing the eigenvalues directly, we recall their relationship to the trace and determinant of the matrix J. Namely, we'll have that the $Re(\lambda_i) > 0$, for $i = 1, 2$ if $\text{trace}(J) > 0$ and $\det(J) > 0$.

For the determinant, one can show that

$$\det(J(u^*, u^*)) = \left(1 + \frac{ad}{(u^* + d)^2}\right) bu^*, \tag{14}$$

and therefore, $\det(J(u^*, u^*)) > 0$. From the condition that the trace be greater than zero, we have

$$u^* \left(\frac{au^*}{(u * +d)^2} - 1 \right) > b. \tag{15}$$

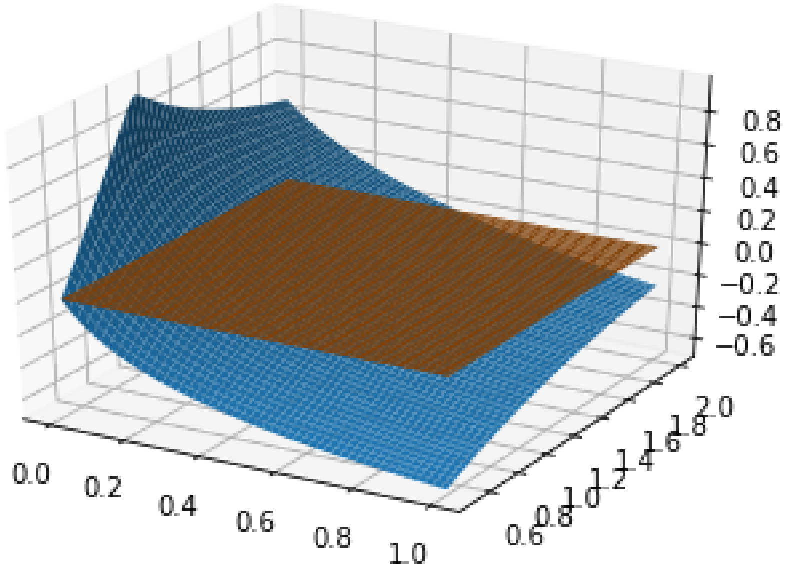
As u^* depends only on the parameters a and d, the equation

$$b(a, d) = u^* \left(\frac{au^*}{(u * +d)^2} - 1 \right) \tag{16}$$

defines a surface in (a, d, b)-space that is the boundary between parameter values that lead to stable fixed points and those that result in unstable fixed points. This surface is plotted below with the region below the surface providing the parameter values for which the fixed point is unstable (Note: we must also have $b > 0$.)

[2]:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

#standard parameter values a = 1, b = 1, d = 0.1

a = np.linspace(0.5,2,200)
d = np.linspace(0.0001,1.0,200)
dmat,amat = np.meshgrid(d,a)
onemamd = 1 - amat - dmat
ustar = 0.5*(onemamd + (onemamd**2 + 4*dmat)**0.5)
bmat = ustar*(amat*ustar/((ustar + dmat)**2) - 1)
b2 = 0*amat;
ax.plot_surface(dmat,amat,bmat,vmin = 0);
ax.plot_surface(dmat,amat,b2);
```

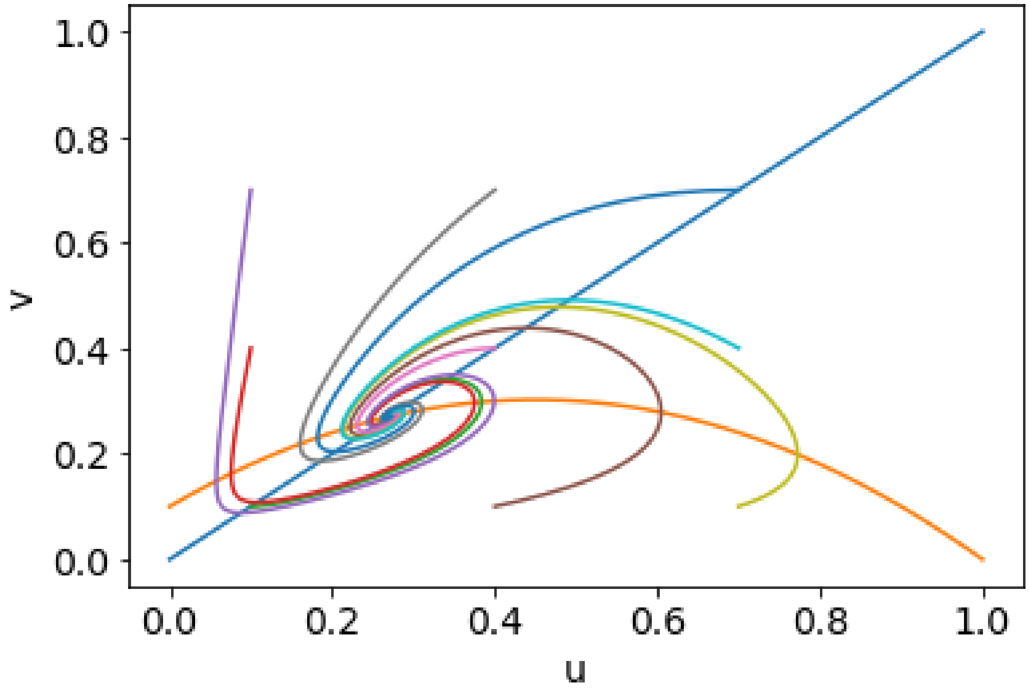


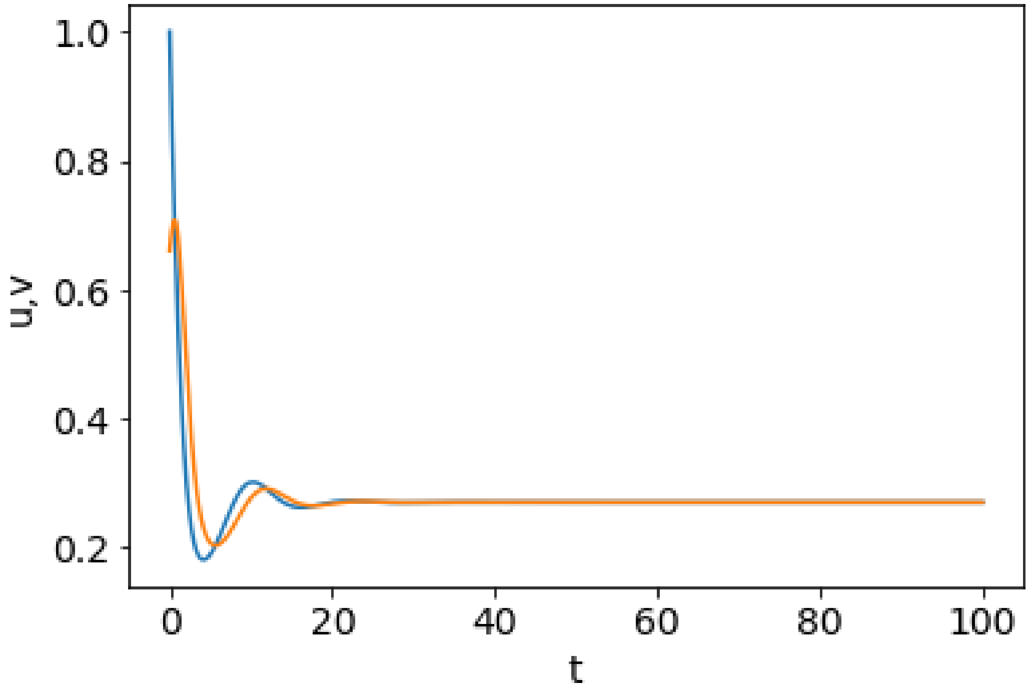
With parameter values taken from below the blue surface, we've satisfied all conditions for Poincaré-Bendixson theorem to hold and we can be guaranteed of the existence of a closed orbit in our trapping region. So what does the solution look like when this occurs? The Python cell below solves the system numerically and shows the trajectories in phase space approaching the limit cycle. The second figure plots u and v as a function of time.

[28]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint



a = 1
b = 0.7
d = 0.1
h = 0.0;

u = np.linspace(0,1,1000)
v1 = u
v2 = (1 - u)*(u + d)/a
plt.figure(0)
plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("u")
plt.ylabel("v")
plt.plot(u,v1)
plt.plot(u,v2)
```

```
def du_dt(u, t):
    return [u[0]*(1 - u[0]) - a*u[0]*u[1]/(u[0] + d) - h, b*u[1]*(1 - u[1]/
 ↪u[0])]

ts = np.linspace(0, 100, 1000)
ics = np.arange(0.1, 0.9, 0.3)
for i in ics:
    for j in ics:
        u0 = [i, j]
        us = odeint(du_dt, u0, ts)
        pop1 = us[:,0]
        pop2 = us[:,1]
        plt.figure(0)
        plt.plot(pop1,pop2);

plt.xlabel("u");
plt.ylabel("v");

u0 = [1.0, 0.66]
us = odeint(du_dt, u0, ts)
pop1 = us[:,0]
pop2 = us[:,1]
plt.figure(1)
plt.plot(ts,pop1)
plt.plot(ts,pop2);
plt.xlabel("t");
plt.ylabel("u,v");
```





You can see immediately the qualitative difference between the outcome of this model and that of the Lotka-Volterra system. For this set of parameter values, all initial conditions approach a single closed orbit in phase space – the limit cycle. This indicates that there is a *natural* oscillatory behaviour of the system that is independent of the intial conditions. Perhaps this is what Volterra was after! It's worth running the cell with different parameter values to confirm the results of our analysis.

One interesting mathematical aspect to point out is that in order for our limit cycle to exist, the fixed point needed to be an unstable node or unstable spiral and we found the parameter values for which this occurs. We can imagine then allowing the parameters to vary and cross over from the region where the fixed point is stable to where it is unstable. When we cross the boundary the change in stability indicates that a bifurcation has occured, though this is not like any we have seen before. At the bifurcation, a stable limit cycle emerges from the fixed point. This is known as a *supercritical Hopf bifurcation* which we'll soon examine in more detail.

1.2 Relaxation oscillators

In our example, the Poincaré-Bendixson theorem proved itself useful in helping us determine whether a closed orbit exists or not, but cannot do much more than that, other than generally indicate where in phase space an orbit might be. Typically, we'd like to be able to estimate the radius and shape of the orbit, as well as its period – the time required for the solution to go once around orbit. We'll see how we can ascertain these details under certain conditions or assumptions.

The first case that we'll consider is that of a *relaxation oscillator*. Let's recall for the moment something that we encountered when studying bifurcations in one-dimensional systems. We saw that certain systems, such as the spruce budworm model, can exhibit hysteresis, where the continuous variation of a parameter caused the fixed point to suddenly 'jump.' In the case of the spruce budworm, this entailed suddenly jumping from the refuge population level to the outbreak level as we increased the parameter R. If we then allowed the parameter to return to its original value, the fixed point did not return to its original location. Recall that for the system to exhibit hysteresis, we needed to have a region of bistability where two stable fixed points exist for a range of parameter values.

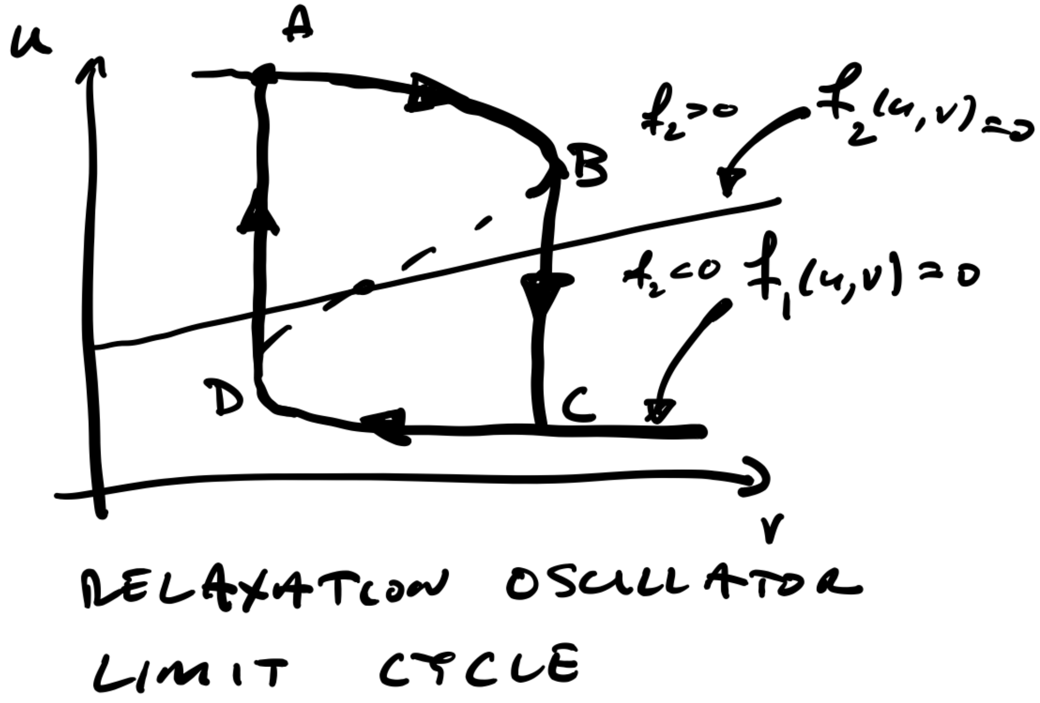
Now suppose we had a one-dimensional system

$$\frac{du}{dt} = f_1(u; v) \tag{17}$$

that exihibits bistability for a range of parameter v. Now's let's turn the parameter into another variable and consider the system

$$\begin{aligned} \frac{du}{dt} &= f_1(u, v) \qquad (18) \\ \frac{dv}{dt} &= \epsilon f_2(u, v) \qquad (19) \end{aligned}$$

If $0 < \epsilon \ll 1$, then we have that $dv/dt \approx 0$ and we can consider it constant while u evolves in time and reaches its steady-state value based on $f_1(u, v) = 0$ for fixed v. Thus, the solution will follow the nullcline $f_1(u, v) = 0$. The solution moves along the nullcline until it reaches a local maximum or minimum, at which point it leaves the nullcline and moves rapidly across phase space at a nearly constant value of v. The point at which it leaves the nullcline would correspond to the bifurcation point if v were a parameter in the one-dimensional system. When its path reaches the nullcline, it moves along it again and the process repeats. This is depicted in the sketch below.



Something else to notice is that we cannot completely ignore what is happening with $f_2(u, v)$. This will give us information about how v varies over a longer timescale, and accordingly which direction the solution moves along the $f_1(u, v) = 0$ nullcline.

Since u evolves rapidly, we refer to it as the *fast variable*, while v is called the *slow variable*. Does this look familiar? We encountered fast and slow variables previously when studying the Michaelis-Menten model of enzyme dynamics (can you identify the fast and slow variables in this case?). The Michaelis-Menten model, however, does not emit oscillatory solutions since the one-dimensional system for the fast variable does not exhibit hysteresis with respect to variations of the slow variable, i.e. the corresponding nullcline $f_1(u, v) = 0$ does not have the right shape.

Let's see how this plays out in an important model of the electrical signals along neurons (nerve cells), the Fitzhugh-Nagumo model.

1.2.1 Fitzhugh-Nagumo model

Before getting into the specifics of the Fitzhugh-Nagumo model, it must be noted that this model is a model of another model, the Hodgkin-Huxley model (the Huxley of note here is Andrew Huxley, grandson of our building's namesake). Hodgkin and Huxley measured the current across the membrane surrounding the axon (the slender appendage eminating from a nerve cell along which electrical signal propagate) due to the passage of different ions through its membrane. In particular, they measured how membrane permeability differs for the different ions and how all permeabilities (ion conductances) depend on the voltage across the membrane. Hodgkin and Huxley were awarded the Nobel Prize for this work and developed four-dimensional system of equations based on their measurements. Unfortunately, the resulting system is rather complex leading researchers to develop simpler models that capture the key features of neuron firing, but are more amenable to analysis. This is where the Fitzhugh-Nagumo models arrives.

The (dimensionless) Fitzhugh-Nagumo model is

$$\frac{dv}{dt} = f(v) - w + I_a \tag{20}$$

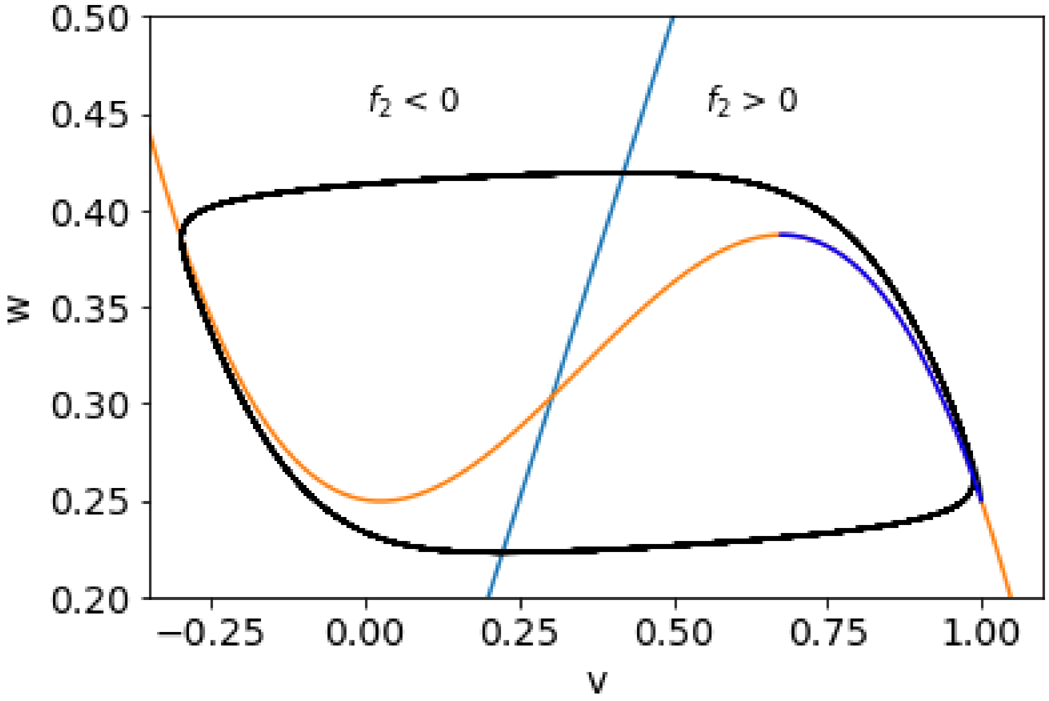
$$\frac{dw}{dt} = bv - \gamma w \tag{21}$$

where $f(v) = v(a - v)(v - 1)$, the variable v is meant to described the dynamics of the membrane voltage and w represents the effect of ion conductance. Additionally, we have $0 < a < 1$ and b and γ are positive constants, while I_a is a current that can be applied across the membrane.

A nice feature of this model is the diversity of the kinds of solutions that one can obtain (that is code for there are some nice bifurcations to observe), including relaxation oscillations. To obtain relaxation oscillations with the Fitzhugh-Nagumo model, we'll take $b = \gamma << 1$, such that $dw/dt \approx 0$ and consequently, v will be the fast variable, while w will be the slow variable. Additionally, for relaxation oscillations to occur, we'll need to set the I_a to ensure that at the fixed point we'll have $df/dv > 0$. Such a case is plotted below using the python cell, taking $a = 0.05$, $b = \gamma = 0.001$, and $I_a = 0.25$.

[7]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint

#Set the parameter values
a = 0.05
b = 0.01
gam = b
Ia = 0.25

#Compute the nullclines
v = np.linspace(-0.5,1.1,100)
w1 = v
fv = v*(a - v)*(v - 1)
w2 = fv + Ia

#Plot the nullclines
plt.rcParams.update({'font.size': 14})  # increase the font size
plt.xlabel("v")
plt.ylabel("w")
plt.plot(v,w1)
plt.plot(v,w2)
plt.ylim(0.2,0.5)
plt.xlim(-0.35,1.1);

plt.text(0.0, 0.45, "$f_2$ < 0", fontsize=12);
plt.text(0.55, 0.45, "$f_2$ > 0", fontsize=12);

#Compute and plot numerical solution to the system
def du_dt(u, t):
    return [u[0]*(a - u[0])*(u[0] - 1) - u[1] + Ia, b*u[0] - gam*u[1]]

ts = np.linspace(0, 1000, 10000)
u0 = [1, Ia]
us = odeint(du_dt, u0, ts)
vvec = us[:,0]
wvec = us[:,1]
plt.figure(1)
plt.plot(vvec,wvec,'k-')
plt.xlabel("v");
plt.ylabel("w");

#Plot section of nullcline used to compute the period
vp = 2/3 + a/6
Nv = 1000
vtest = np.linspace(1, vp, Nv)
wtest = vtest*(a - vtest)*(vtest - 1) + Ia
plt.plot(vtest,wtest,'b-');
```



For this set of parameters, since $b = \gamma \ll 1$, the solution will evolve more or less along the nullcline given by $f_1(v,w) = 0$. Since we have $f_2 < 0$ to the left and above the nullcline $f_2(v,w) = 0$, the solution will move downward along nullcline $f_1(v,w) = 0$ in this region until it comes close to the local minimum $df/dv = 0$ at $v \approx 0$. At this point, the solution moves rapidly to the right across phase space to again meet the nullcline $v \approx 1$. Since here $f_2 > 0$, we will move up the curve to the local maximum, at which point the solution will move rapidly to the left to meet the nullcline at $v \approx -0.3$ and the process repeats, completing the cycle. This behaviour is confirmed by the numerical solution (black curve) shown in the plot above. It's interesting to see how the orbit changes shape as you increase the value of b and γ – try it!

Period of the orbit In addition to being able predict the shape of the limit cycle in phase space for a relaxation oscillator, we are also able to approximate the period of the orbit. Recall that for a relaxation oscillator, there are the fast and slow variables, here, v and w, respectively. The equation for the fast variable tells us which nullcline the solution will move along, while the equation for the slow variable tells us the direction the solution will move.

As the evolution is governed by the change in the slow variable, to approximate the period of the orbit, we only need to consider places where w is changing. Thus, since the 'jumps' in the solution that occur near the local maxima and minima of the nullcline have $w \approx$ constant, they can be considered to occur instantaneously and only motion along the nullcline will contribute to the time required to make one orbit.

With the above in mind, we first determine the points in vw-plane where the local minima and maxima of the nullcline occur. To find these points, we must consider $df/dv = -3v^2+2(1+a)v-a = 0$, which yields,

$$v_{\pm} = \frac{1}{3}\left((1+a) \pm \sqrt{1-a+a^2}\right). \tag{22}$$

To get a better idea of these values, if we assume that $a \ll 1$, we'll have $v_+ = 2/3 + a/6 + O(a^2)$ and $v_- = a/2 + O(a^2)$, indicating that at least in this limit, the roots are real. For v_-, we can find the corresponding value of w_-, by evaluating our expression for the nullcline, $w_- = f(v_-) + I_a$. In the limit $a \ll 1$, we'll have $w_- = I_a + O(a^2)$. Taking the line $w = w_-$, we find where it will intersect the nullcline again by considering $w_- = f(v_*) + I_a$. In the $a \ll 1$, the values is $v_* = 1 + O(a^2)$. Therefore, what we'd like to compute is the time required for the solution to move from $v = v_*$ to $v = v_+$ along the nullcline. This approximate path is the blue curve shown in the plot above containing the nullclines and solution. We now can find its contribution to the period.

By integrating the differential equation for the slow variable, we have

$$\int_{w(v_*)}^{w(v_+)} \frac{dw}{v-w} = \int_0^{T_1} \gamma dt \tag{23}$$

Since along the nullcline $w = f(v) + I_a$ and therefore $dw/dv = df/dv$, we can rewrite the integral on the lefthand side as

$$\int_{v_*}^{v_+} \frac{1}{v - f(v) - I_a}\frac{dw}{dv}dv = \int_1^{v_+} \frac{-3v^2 + 2(a+1)v - a}{v - f(v) - I_a}dv \tag{24}$$

and thus the time required to move along this part of the orbit is

$$T_1 = \frac{1}{\gamma}\int_1^{v_+} \frac{-3v^2 + 2(a+1)v - a}{v - f(v) - I_a}dv. \tag{25}$$

Similarly, we can compute the contribution to the period from the other part of the orbit as

$$T_2 = \frac{1}{\gamma}\int_{v_+ - 1 + a/2}^{v_-} \frac{-3v^2 + 2(a+1)v - a}{v - f(v) - I_a}dv, \tag{26}$$

giving the total period as $T = T_1 + T_2$.

While these expressions might be a bit unwieldy to be evaluated analytically, we can apply basic quadrature to obtain a numerical value as is done in the Python cell below.

[3]:

```python
import numpy as np
import matplotlib.pyplot as plt

#Set the parameters
```

```
a = 0.05
b = 0.001
gam = b
Ia = 0.25

#Integrate numerically to obtain T_1
vp = 2/3 + a/6
vm = a/2
vstarstar = vp - 1 + vm

Nv = 1000
v = np.linspace(1, vp, Nv)
dv = (1 - vp)/Nv
fnum = -3*v**2 + 2*(a + 1)*v - a
fden = v - v*(a - v)*(v - 1) - Ia
f = fnum/fden
Integral1 = sum(f)*dv
T1 = -Integral1/gam

#Integrate numerically to obtain T_2
Nv = 1000
v = np.linspace(vstarstar, vm, Nv)
dv = (vm - vstarstar)/Nv
fnum = -3*v**2 + 2*(a + 1)*v - a
fden = v - v*(a - v)*(v - 1) - Ia
f = fnum/fden
Integral2 = sum(f)*dv
T2 = Integral2/gam
T = T1 + T2
print("The approximate period of the limit cycle is", T)
```

```
The approximate period of the limit cycle is 528.396812249234
```

```
[ ]:
```

Mathematical Biology - Week 6

November 1, 2022

0.1 Bifurcations

An important aspect of one-dimensional systems was bifurcations – the qualitative changes in the fixed points that occur when parameters in the model were varied. We observed that bifurcations can have important biological implications, such as whether disease outbreaks occur and possibly remain due to hysteresis.

In our discussion of multidimensional systems, we've briefly mentioned or even only hinted at the kinds of bifurcations that were occuring in some of the models that we've studied. We'll now present bifurcations in multidimensional systems in a bit more detail, focusing on the case where $n = 2$. We'll examine closely Hopf bifurcations and tie this in with our study of closed orbits and limit cycles. This will broaden our definition of a bifurcation in that they no longer apply solely to fixed points, but involve other kinds of attractors (stable/unstable limit cycles) as well.

0.1.1 Saddle-node, transcritical and pitchfork bifurcations

In our discussion of one-dimensional systems, we encountered the 3 different types of bifurcation that occur and described them using their normal forms. In two dimensions, these types of bifurcations are broadly similar in that when they occur, the motion of the fixed points is restricted to a one-dimensional subspace of the phase plane, to which the flow is attacted or repelled. To see this, we can characterise these three types of bifurcations in 2D through the following simple examples where the one-dimensional subspace is the x-axis.

A simple example of the *saddle-node bifurcation* is

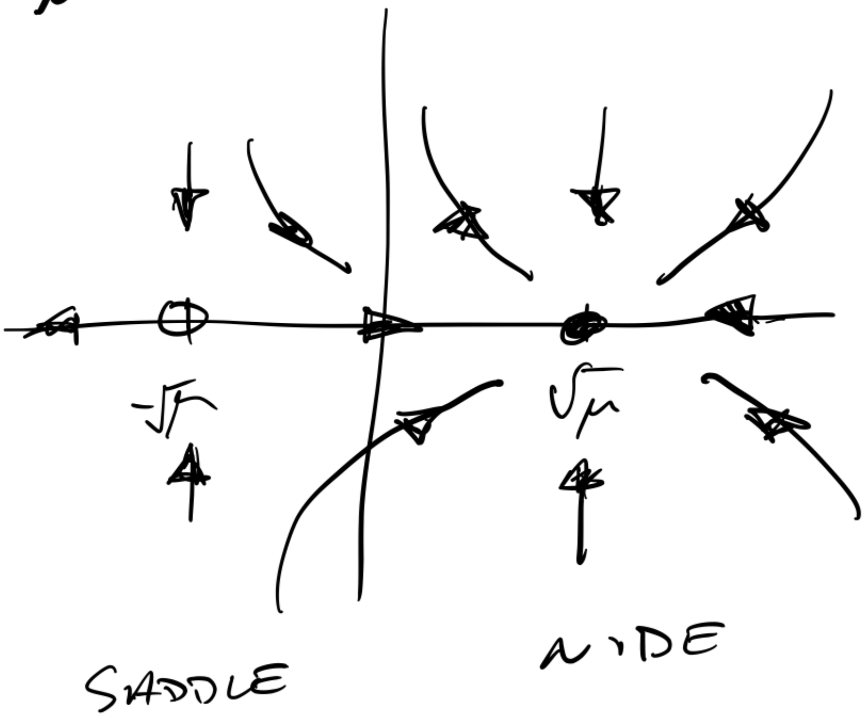
$$\frac{dx}{dt} = \mu - x^2, \tag{1}$$

$$\frac{dy}{dt} = -y. \tag{2}$$

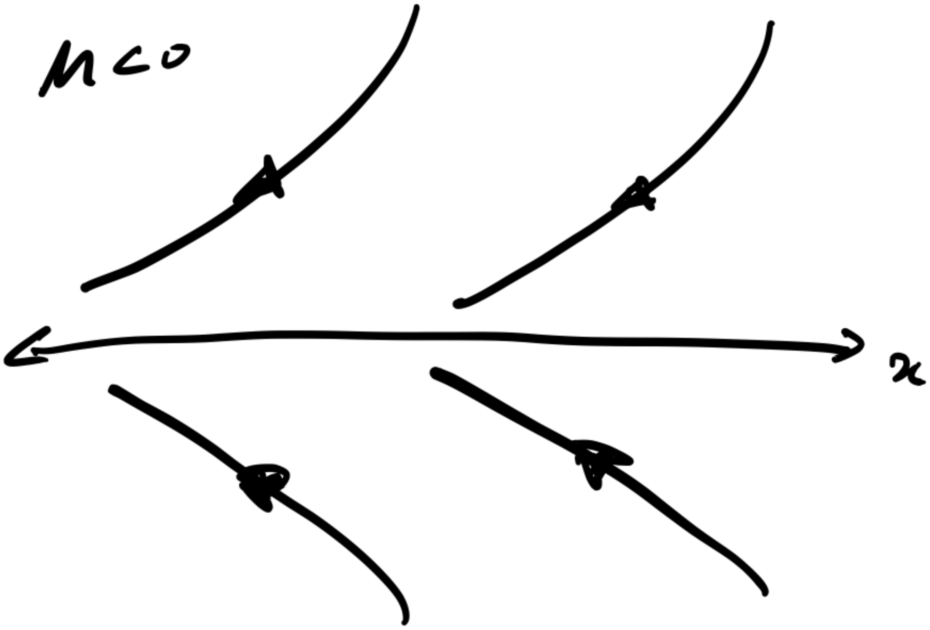
The first equation above is exactly the same as that encountered in one dimensional systems, while the second describes motion toward the x-axis. As we saw with one-dimensional systems, for $\mu > 0$, there are 2 fixed points at $(\pm\sqrt{\mu}, 0)$, for $\mu = 0$ there is a single fixed point at $(0, 0)$, and for $\mu < 0$ there are no fixed points. The flow for these different cases is depicted below.

SADDLE-NODE BIFURCATION

$\mu > 0$



$\mu < 0$



NO FIXED POINTS.

In the previous section, we considered the genetic control system

$$\frac{du}{d\tau} = av - bu, \tag{3}$$
$$\frac{dv}{d\tau} = \frac{u^2}{1+u^2} - v, \tag{4}$$

we found that if $a/b < 1/2$ there were 3 fixed points, while if $a/b > 1/2$, there was a single point at the origin. When $a/b = 1/2$, we observed the emergence of a saddle and node pair, indicating the occurence of the saddle-node bifurcation. We see that the one-dimensional subspace along which the fixed points move is not as simple as the x-axis in our example, but we see that, at least locally near the saddle, it can be described by the unstable manifold described by the eigenvector associated with the unstable eigenvalue.

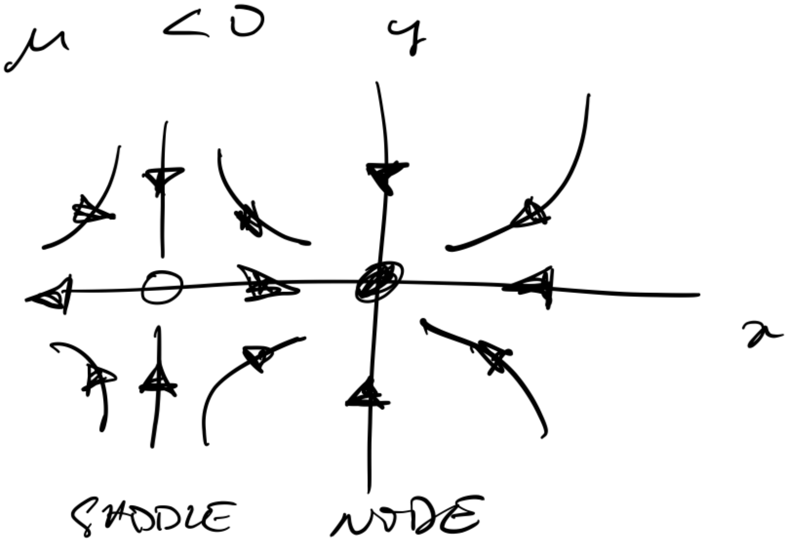
For the *transcritical* bifurcation, we have

$$\frac{dx}{dt} = \mu x - x^2, \tag{5}$$
$$\frac{dy}{dt} = -y, \tag{6}$$

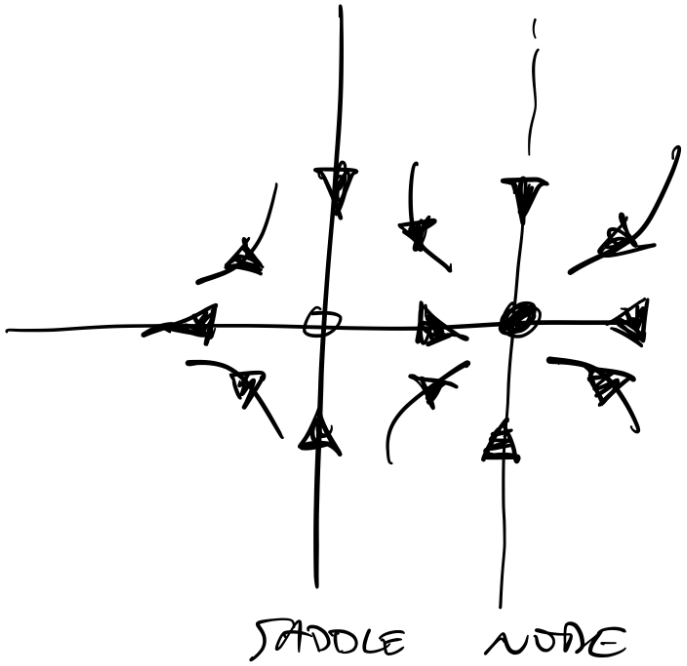
where for $\mu < 0$ we will have a saddle at $(\mu, 0)$ and a stable node at $(0, 0)$. For $\mu > 0$, the fixed point $(0, 0)$ is a saddle, while at $(\mu, 0)$ we have a stable node. This is illustrated in the sketch below.

TRANSCRITICAL BIFURCATION

$\mu < 0$



$\mu > 0$



Finally, we have the *supercritcal pitchfork*,

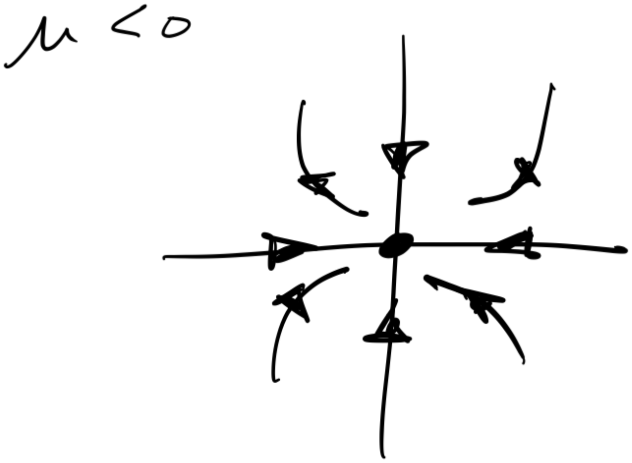
$$\frac{dx}{dt} = \mu x - x^3, \tag{7}$$
$$\frac{dy}{dt} = -y, \tag{8}$$

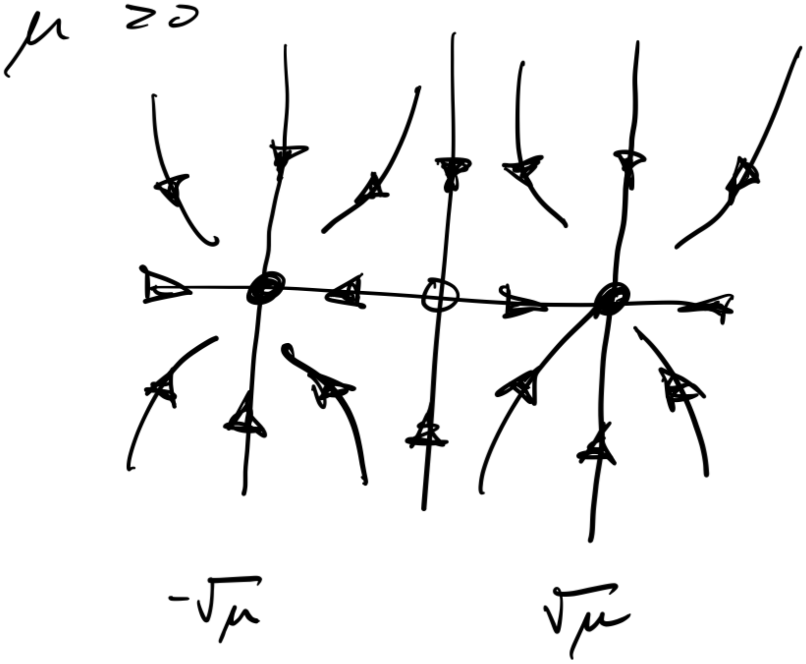
as well as the *subcritical pitchfork*

$$\frac{dx}{dt} = \mu x + x^3, \tag{9}$$
$$\frac{dy}{dt} = -y. \tag{10}$$

Again, very much like in one dimension, for the supercritical case, we have a single stable node at the origin if $\mu < 0$. When $\mu > 0$, the node at the origin is a saddle, while at $(\pm\sqrt{\mu}, 0)$ we will have stable nodes. The flow is sketched below.

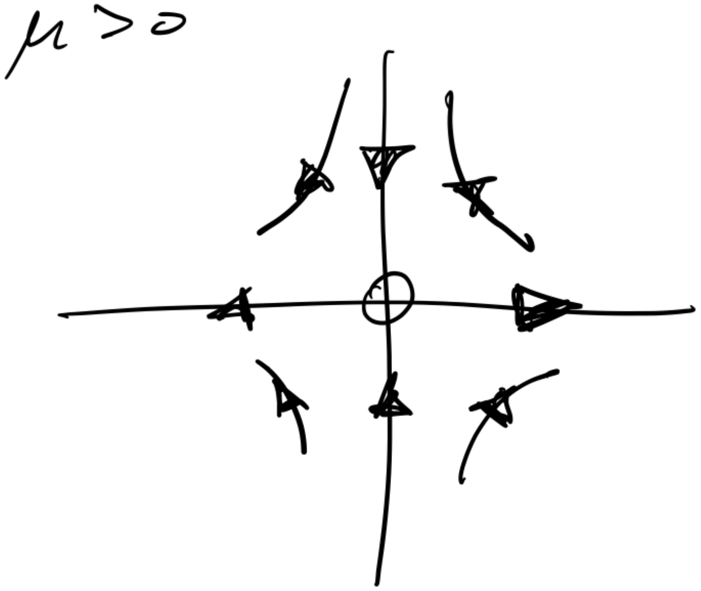
SUPERCRITICAL PITCHFORK

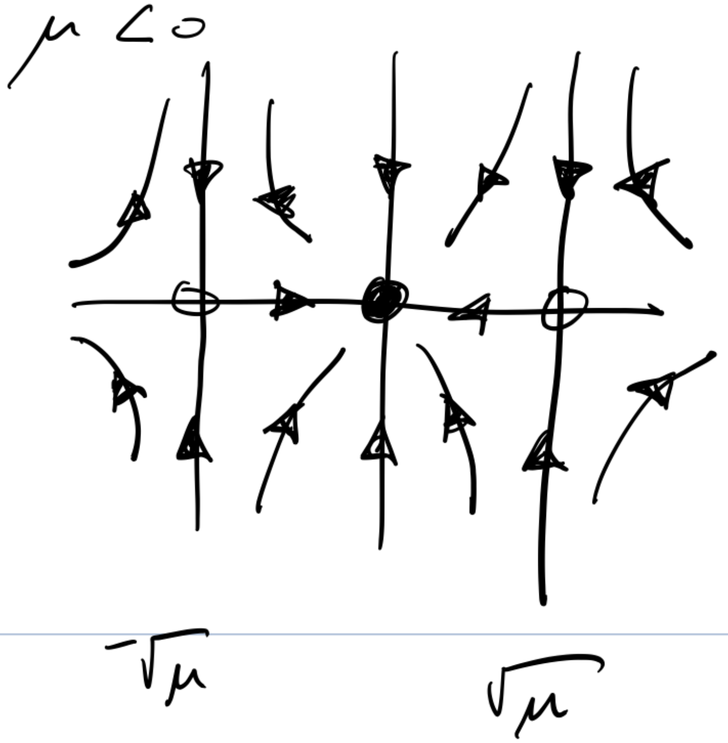




For the subcritical pitchfork, we'll have a saddle at the origin if $\mu > 0$. When $\mu < 0$, the saddle at the origin becomes a stable node, while at $(\pm\sqrt{\mu}, 0)$ we will have saddles. The flow is sketched below.

SUBCRITICAL PITCHFORK





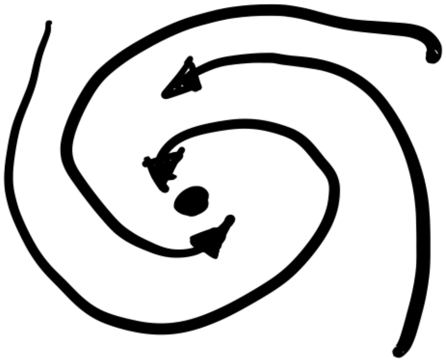
For each type of bifurcation – saddle-node, transcritical and pitchfork – we see that the normal forms for the one-dimensional cases are extended to two dimensions simply by including the addtional equation $dy/dt = -y$ which requires the flow to approach the x-axis. More generally, as we saw in the case of the genetic control system, the one-dimensional curve along which the fixed points move will not be the x-axis, or even a straight line. Just as in the case of the normal forms, these examples are meant to give a simple local description of the bifurcation with the x-axis coinciding with the unstable manifold eminating from the saddles very close to the parameter values where the bifurcation occurs.

0.2 Hopf Bifurcations

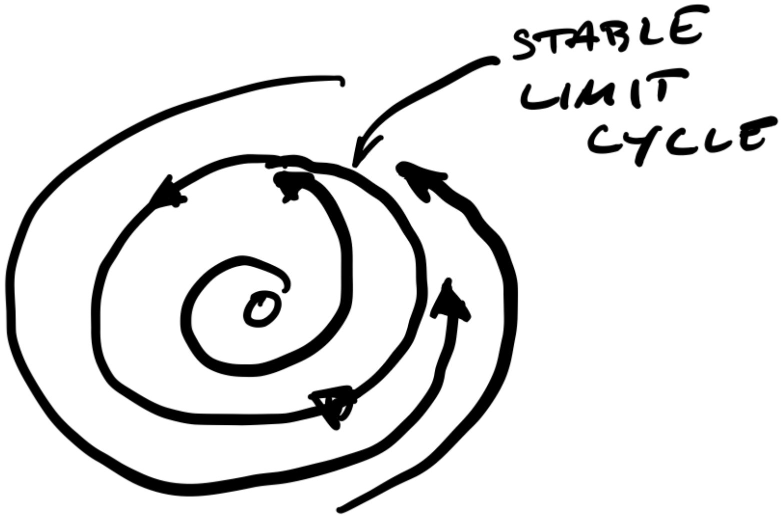
While the bifurcations described above are very similar to what we have encountered in one dimension, the Hopf bifurcation we will discuss now is entirely new as it involves closed orbits. Like our friend the pitchfork bifurcation, the Hopf bifurcations comes in the super- and subcritical flavours. The supercritical Hopf bifurcation involves a stable spiral becoming unstable while simultaneously a limit cycle emerges from the fixed point. In the subcritical case, a stable spiral surrounded by an unstable cycle becomes unstable when the unstable cycle collapses in on the fixed point at the bifurcation. These two cases are depicted below.

SUPERCRITICAL HOPF BIFURCATION

$\mu < \mu_c$



$\mu > \mu_c$



Consider the 2D system

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; \mu), \tag{11}$$

where μ is a parameter. Suppose that system has a fixed point $\mathbf{x}^*(\mu)$ which is a spiral. Thus, we can express the eigenvalues of the Jacobian as $\lambda(\mu) = \alpha(\mu) \pm i\omega(\mu)$. Suppose also that at $\mu = \mu_c$ a Hopf bifurcation occurs, where the fixed point changes from being a stable spiral to one that is unstable. Thus, at the μ_c, we must have $\alpha(\mu_c) = 0$.

We'll now attempt to gain some insight into the limit cycle that emerges at the Hopf bifurcation. This involves what is known as a *weakly nonlinear analysis* of the system. It entails a multiple scales analysis that yields the equations for the limit cycle near the bifurcation.

This analysis will be one of the more technically involved aspects of the module. I'll attempt to describe the key steps of the analysis and discuss why they are taken. For those of you that are interested in the more gory details, have a look at the book *Chemical oscillations, waves, and turbulence* by Y. Kuramoto.

1. The first step is to expand the function $\mathbf{f}(\mathbf{x};\mu)$ to *cubic order* about the fixed point $\mathbf{x}^*$, such that

   $$\frac{d\mathbf{X}}{dt} = J\mathbf{X} + M\mathbf{X}\mathbf{X} + N\mathbf{X}\mathbf{X}\mathbf{X} + ... \tag{12}$$

   where $\mathbf{X} = \mathbf{x} - \mathbf{x}^*$ and, as usual, J is the Jacobian evaluated at the fixed point. The entries of the other tensors are related to the higher-order partial derivatives of $\mathbf{f}$ through

   $$M_{ijk} = \frac{1}{2}\frac{\partial^2 f_i(\mathbf{x}^*)}{\partial x_j \partial x_k} \tag{13}$$
   $$N_{ijkl} = \frac{1}{6}\frac{\partial^3 f_i(\mathbf{x}^*)}{\partial x_j \partial x_k \partial x_l}. \tag{14}$$

   This is identical to what we've done so far with when we've performed linear analysis, but now we're retaining terms beyond the Jacobian (hence the name weakly nonlinear analysis).

2. We've expanded about $\mathbf{x}^*$. The next step is to expand J, M, and N in the bifurcation parameter such that

   $$\begin{aligned} J &= J(\mu_c) + \eta\frac{dJ}{d\mu}(\mu_c) = J_0 + \eta J_1 + ... \\ M &= M(\mu_c) + \eta\frac{dM}{d\mu}(\mu_c) = M_0 + \eta M_1 + ... \\ N &= N(\mu_c) + \eta\frac{dN}{d\mu}(\mu_c) = N_0 + \eta N_1 + ... \end{aligned} \tag{15}$$

   where $\eta = \mu - \mu_c$ and $|\eta| = \epsilon \ll 1$.

   Let's examine the Jacobian for a moment. The first term is $J_0 = J(\mu_c)$, for which we have $\lambda_0 = \pm i\omega(\mu_c)$. Thus, for J_1 we must have $\lambda_1 = \alpha_1 \pm i\omega_1$ with $\alpha_1 > 0$ in order to have the unstable spiral.

3. The next thing to notice is that the expansion of the Jacobian generates two timescales: a fast timescale for the oscillations given by $T_f \sim \omega(\mu_c)^{-1}$ and slow timescale that is due to the growth of the spiral, $T_s \sim \epsilon^{-1}$. This information will be used in the multiple scales analysis, referred to as two-timing by Strogatz. In doing so, time is split between two variables, t, and slow time $\tau = \epsilon t$. As a result, we have

$$\frac{d}{dt} \to \frac{\partial}{\partial t} + \epsilon \frac{\partial}{\partial \tau}, \tag{16}$$

turning our ODE into a PDE.

4. A final step is to expand $\mathbf{X}(t,\tau)$ such that $\mathbf{X}(t,\tau) = \epsilon^{1/2}\mathbf{X}_1(t,\tau) + \epsilon\mathbf{X}_2(t,\tau) + \epsilon^{3/2}\mathbf{X}_3(t,\tau) + \dots$. The choice of the expansion in 1/2-powers of ϵ ensures that the growth of the solution due to ηJ_1 can be controlled by the cubic term in the expansion. What this means is that the spiral will not keep spiralling and instead approach a limit cycle.

5. Now the expansions are substituted into (11) and we require that the terms balance at each order $\epsilon^{n/2}$ for $n = 1, \dots$. At order $\epsilon^{1/2}$, we have

$$\frac{\partial \mathbf{X}_1}{\partial t} = J_0 \mathbf{X}_1 \tag{17}$$

which has solution

$$\mathbf{X}_1(t,\tau) = A(\tau)\boldsymbol{\xi}\mathrm{e}^{i\omega(\mu_c)t} + \overline{A(\tau)\boldsymbol{\xi}}\mathrm{e}^{-i\omega(\mu_c)t} \tag{18}$$

where $\boldsymbol{\xi}$ is the eigenvector associated with eigenvalue $i\omega(\mu_c)$, $A(\tau)$ is the complex amplitude that evolves with the slow time, and the overbar denotes the complex conjugate.

6. Obtaining the equation for A involves considering the equations that arise at higher-order in the expansion. It is not until order $\epsilon^{3/2}$ that you obtain (here is where many gory details are skipped) the equation

$$\frac{dA}{d\tau} = \frac{\eta}{\epsilon}\alpha_1 A - g|A|^2 A \tag{19}$$

where $g = g' + ig''$ is a complex number that depends on M_0, N_0, among other things (we'll discuss this in a moment). This equation is commonly known as the Stuart-Landau equation, named, in part after J. Trevor Stuart, Professor Emeritus of our department.

To understand (19) equation, it's useful to introduce polar coordinates and write $A(\tau) = R(\tau)\exp(i\theta(\tau))$. Thus, R is related the radius of the limit cycle. The resulting equations for R and θ are

$$\frac{dR}{d\tau} = \frac{\eta}{\epsilon}\alpha_1 R - g' R^3 \tag{20}$$

$$\frac{d\theta}{d\tau} = \frac{\eta}{\epsilon}\omega_1 - g'' R^2 \tag{21}$$

Have a look at the equation for R. Does it look somewhat familiar? It should – it's very close (in fact, could easily be transformed) to the normal form for a pitchfork bifurcation! Thus, the Hopf bifurcation can be viewed simply as a pitchfork bifurcation for the radius of the limit cycle.

For $g' > 0$, we have the nontrivial fixed point for the R equation

$$R_s = \sqrt{\alpha_1/|g'|} \tag{22}$$

when $\eta/\epsilon = 1$ with $\theta = \Omega\tau + \text{const.}$, where $\Omega = \omega_1 - g'' R_s^2$. This corresponds to the supercritical case and R_s gives the approximate radius of the stable limit cycle. For $g' < 0$ and $\eta/\epsilon = -1$, we have $R = R_s$ and $\theta = -\Omega\tau + \text{const.}$. This is the subcritical case, and the cycle corresponding to R_s is unstable. Using this information in the expression for $\mathbf{X}_1$, (18), we have the limit cycle solution

$$\mathbf{x}(t) = \mathbf{x}^* + \epsilon^{1/2}\mathbf{X}_1 = \mathbf{x}^* + \epsilon^{1/2} R_s \left(\boldsymbol{\xi} \mathrm{e}^{i(\omega(\mu_c)+\epsilon\Omega)t} + \overline{\boldsymbol{\xi}} \mathrm{e}^{-i(\omega(\mu_c)+\epsilon\Omega)t} \right). \tag{23}$$

We see then that to order ϵ, the period of the cycle is $T = 2\pi/\omega(\mu_c)$. Thus, near the bifurcation, the period of the cycle is given purely by the eigenvalues of the matrix J_0.

The final thing to introduce is the rather complicated expression for the complex number g that pops up in the equations for A. This factor is related to the solution $\mathbf{X}_2$ that comes in at order ϵ and affects the solution at order $\epsilon^{3/2}$.

The expression for g taken from Kuramoto's book is

$$g = \overline{\boldsymbol{\xi}} \cdot \left(-2M_0 \boldsymbol{\xi}\boldsymbol{\zeta}_1 - 2M_0 \overline{\boldsymbol{\xi}}\boldsymbol{\zeta}_2 - 3N_0 \boldsymbol{\xi}\boldsymbol{\xi}\overline{\boldsymbol{\xi}} \right) \tag{24}$$

where

$$\boldsymbol{\zeta}_1 = -2J_0^{-1} M_0 \boldsymbol{\xi}\overline{\boldsymbol{\xi}} \tag{25}$$

$$\boldsymbol{\zeta}_2 = -(J_0 - 2i\omega(\mu_c))^{-1} M_0 \boldsymbol{\xi}\boldsymbol{\xi}. \tag{26}$$

0.2.1 Hopf bifurcation in the Fitzhugh-Nagumo model

Let's apply this theory to the Fitzhugh-Nagumo model

$$\frac{dv}{dt} = f(v) - w + I_a \tag{27}$$
$$\frac{dw}{dt} = bv - \gamma w \tag{28}$$

where, if you recall, $f(v) = v(a-v)(v-1)$. For simplicity, we'll take the case where $b = \gamma$. The bifurcation parameter will be the applied current I_a. Below a critical value of I_a, we'll have a stable fixed point and the membrane voltage will remain constant. Once the applied current is sufficiently high, we will have a Hopf bifurcation and the membrane voltage will begin to oscillate. We'd like to quantify these oscillation when the applied current is just above the critical value.

The fixed point and applied current at the bifurcation The first step is to determine the fixed point and the applied current when the bifurcations occurs. We begin by computing the Jacobian,

$$J(v,w) = \begin{bmatrix} df/dv & -1 \\ \gamma & -\gamma \end{bmatrix} \tag{29}$$

and its eigenvalues

$$\lambda_\pm = \frac{1}{2}\left(-(\gamma - df/dv) \pm \sqrt{(\gamma - df/dv)^2 - 4\gamma(1 - df/dv)}\right). \tag{30}$$

It may seem like we are doing things out of order (usually, we evaluate J at the fixed points before finding the eigenvalues), but for now, let's just assume that df/dv is evaluated at the fixed point.

When a Hopf bifurcation occurs, the fixed point will change from being a stable spiral to an unstable spiral. Thus, at the bifurcation we must have $Re(\lambda_\pm) = 0$ and therefore we can write $\lambda_\pm = \pm i\omega$. We will refer to ω as the frequency.

For the Fitzhugh-Nagumo model, this means that at the Hopf bifurcation we'll have

$$\frac{df}{dv} = \gamma \tag{31}$$

and $(\gamma - df/dv)^2 - 4\gamma(1 - df/dv) < 0$, which combined with the fact that $df/dv = \gamma$, gives

$$-4\gamma(1-\gamma) < 0. \tag{32}$$

Thus, we must also have $\gamma < 1$. If this is satisified, the frequency is given by $\omega^2 = \gamma(1-\gamma)$.

We can use the condition $df/dv = \gamma$ at the bifurcation to find the fixed point at the bifurcation. Accordingly, the fixed point must satisfy

$$\gamma = -3v^2 + 2(1+a)v - a \tag{33}$$

which gives

$$v^* = \frac{1}{3}\left((1+a) \pm \sqrt{(}(1+a)^2 - 3(a+\gamma)\right). \tag{34}$$

The fact that v^* must be real requires that $(1+a)^2 - 3(a+\gamma) > 0$ and hence

$$\frac{1}{3}(1+a)^2 - a > \gamma. \tag{35}$$

Since v^* is a fixed point, it must also satisfy

$$0 = f(v^*) - w^* + I_a \tag{36}$$
$$0 = \gamma v^* - \gamma w^*. \tag{37}$$

Using these equations, we can solve for the value of the current at the bifurcation,

$$I_a^* = v^*(1 - (a - v^*)(v^* - 1)). \tag{38}$$

To quickly recap, the parameter γ must satisfy $(1+a)^2/3 - a > \gamma$ for the bifurcation to occur. Second, the fixed point at the bifurcation is given by (v^*, v^*), where the expression for v^* is provided by (34). At v^* we also have $df/dv = \gamma$ and thus,

$$J_0 = \begin{bmatrix} \gamma & -1 \\ \gamma & -\gamma \end{bmatrix} \tag{39}$$

which has eigenvalues $\lambda = \pm i\omega$ with $\omega^2 = \gamma(1-\gamma)$.

The limit cycle near the bifurcation To obtain an expression for $(v(t), w(t))$ near the bifurcation, we'll need to do a few more things. First, we'll need to find one of the eigenvectors of J_0 by considering

$$J_0 \boldsymbol{\xi} = i\omega\boldsymbol{\xi}. \tag{40}$$

If we also require that $\boldsymbol{\xi}^*\boldsymbol{\xi} = 1$, we have

$$\boldsymbol{\xi} = \begin{bmatrix} \frac{1}{\sqrt{1+\gamma}} \\ \frac{\gamma - i\omega}{\sqrt{1+\gamma}} \end{bmatrix}. \tag{41}$$

Next, we'll need to compute J_1, M_0, and N_0 that appear in (15). We obtain J_1 by expanding the expression for J about I_a^*. This gives

$$J_1 = \frac{dJ}{dI_a} = \begin{bmatrix} \frac{d^2f}{dv^2}\frac{dv}{dI_a} & 0 \\ 0 & 0 \end{bmatrix} \tag{42}$$

where $dv/dI_a = (dI_a/dv)^{-1}$. By differentiating (38) with respect to v, we obtain

$$\frac{dI_a}{dv} = 1 - \frac{df}{dv}(v^*) = 1 - \gamma. \tag{43}$$

Thus, the value of α_1 appearing in (22) is

$$\alpha_1 = (1-\gamma)^{-1}\frac{d^2f}{dv^2}(v^*). \tag{44}$$

The final pieces that we need to compute g are M_0 and N_0. Recall that M_0 and N_0 are related to the second and third partial derivatives, respectively, of f_1 and f_2. Since $f_1(v,w) = f(v) - w + I_a$ depends linearly on w and $f_2(v,w) = \gamma(v-w)$ depends linearly on both v and w, the only nonzero entries of M_0 and N_0 will be

$$M_{0;111} = \frac{1}{2}\frac{d^2f}{dv^2}(v^*) \tag{45}$$

$$N_{0;1111} = \frac{1}{6}\frac{d^3f}{dv^3}(v^*). \tag{46}$$

The Python cell below uses the information that we just obtained to evaluate the expression for g and compute the limit cycle by evaluating (23). For this case and ignoring the contribution of Ω, (23) becomes

$$\mathbf{v} = \mathbf{v}^* + (I_a - I_a^*)^{1/2}R_s\left(\boldsymbol{\xi}\mathrm{e}^{it\sqrt{\gamma(1-\gamma)}} + \bar{\boldsymbol{\xi}}\mathrm{e}^{-it\sqrt{\gamma(1-\gamma)}}\right). \tag{47}$$

where $\mathbf{v} = (v,w)$ and $\mathbf{v}^* = (v^*, v^*)$.

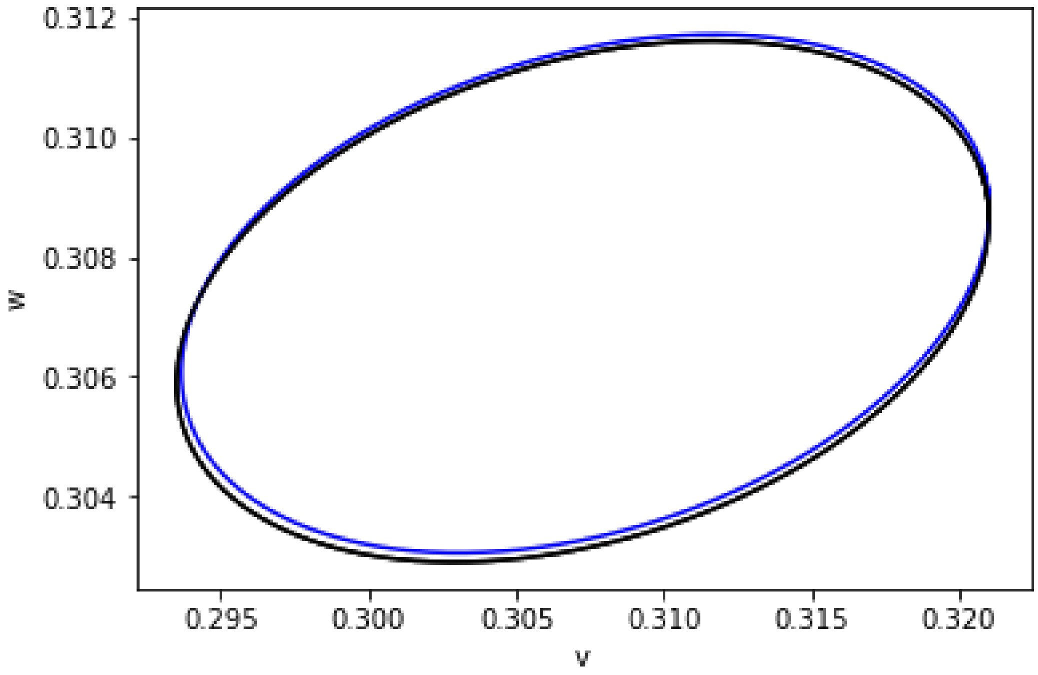
[1]:

```python
import numpy as np
import matplotlib.pyplot as plt
```

```
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

#Set parameters a, b and gamma
a = 0.6
b = 0.1
gam = b

#Compute the value of v and I_a at the bifurcation
vbf = 2*(1+a)-(4*(1+a)**2 - 12*(a+gam))**0.5
vbf = vbf/6
Ia = vbf*(1 - (a - vbf)*(vbf - 1))


#Evaluate the derivative of f and I_a w.r.t. v
d2fdv2 = -6*vbf + 2*(a + 1)
d3fdv3 = -6
dIdv = 1 - gam

#Compute the eigenvalue and eigenvector
omega = (gam*(1 - gam))**0.5
xi = np.array([[1], [gam - 1j*omega]])/(1 + gam)**0.5

#Compute the Jacobian matrices
J0 = np.array([[gam, -1], [gam, -gam]])
J1 = np.array([[d2fdv2/dIdv, 0], [0, 0]])
Id = np.eye(2)

#Compute complex number g
zeta1vec = np.array([0.5*xi[0]*xi[0]*d2fdv2, [0]])
zeta1 = - 2*linalg.solve(J0, zeta1vec)
zeta2 = - linalg.solve(J0 - 2*1j*omega*Id, zeta1vec)
g = xi[0]*(- d2fdv2*xi[0]*zeta1[0] - d2fdv2*xi[0]*zeta2[0] - 0.
 ↪5*d3fdv3*xi[0]**3)

#Find the radius of the limit cycle
Rs = (d2fdv2/(dIdv*g.real))**0.5

#Set the value of (I_a - I_a^*)
dIa = 0.0001*Ia

#Compute the limit cycle
ts = np.linspace(0, 6.5/omega, 1000)
u = np.array([[vbf], [vbf]]) + dIa**0.5*Rs*(xi*np.exp(1j*omega*ts) + np.
 ↪conjugate(xi)*np.exp(-1j*omega*ts))
```

```
#Plot the limit cycle
vvec = u[0,:]
wvec = u[1,:]
plt.figure(1)
plt.plot(vvec,wvec,'k-')
plt.xlabel("v");
plt.ylabel("w");

#Compute and plot the numerical solution of the original system.
def du_dt(usol, t):
    return [usol[0]*(a - usol[0])*(usol[0] - 1) - usol[1] + Ia + dIa, b*usol[0]␣
 ↪- gam*usol[1]]

ts = np.linspace(0, 6.5/omega, 10000)
u0 = [u[0][0].real, u[1][0].real]
us = odeint(du_dt, u0, ts)
vvec2 = us[:,0]
wvec2 = us[:,1]
plt.figure(1)
plt.plot(vvec2,wvec2,'b-')
plt.xlabel("v");
plt.ylabel("w");
plt.plot(vvec,wvec,'k-');
```

```
/Users/ekeaveny/anaconda3/lib/python3.7/site-packages/numpy/core/numeric.py:501:
ComplexWarning: Casting complex values to real discards the imaginary part
  return array(a, dtype, copy=False, order=order)
```



Along with the limit cycle given by (23) which is shown above in black, the solution to the full system obtained by integrating the equations numerically is also provided (blue curve). We see that the approximate solution given by the weakly nonlinear analysis matches quite closely to the numerical solution of the full equations. Recall that a key assumption in finding our approximate solution is that $I_a - I_a^* \ll 1$. Try adjusting this parameter in the cell above and see how the difference between the approximate and numerical solutions change.

We also notice that the limit cycle is not circular, but elliptical. The aspect ratio of the ellipse is determined by the eigenvector, $\boldsymbol{\xi}$. This can be readily observed by converting the expression for the limit cycle in terms of complex numbers to an expression that uses only real quantities.

Finally, we note that we can obtain a very good approximation of the period through $T = 2\pi/\omega = 2\pi/\sqrt{\gamma(1-\gamma)}$. This can be tested by comparing it with the period of the numerical solution.

[ ]:

Mathematical Biology - Week 7

November 16, 2022

1 Spatial dynamics

Thus far, we've used ordinary differential equations to generate mathematical models of population dynamics, biochemical reactions, and epidemiological phenomena, to name a few. In modelling these biological systems in this way, we have implicitly assumed that the phenomena that can occur can be described by functions of time that evolve deterministically. It's not very hard to envision situations, including those we've already discussed, that may also have a spatial dependence. For example, one can easily imagine a population of bacteria or another kind of microorganism that may initially be localised in space, but owing to motility, it may spread over time. As a result, we would not only observe growth in the population size due to cell division, but also in the region of space which the population occupies.

In this part of the course, we'll discuss how ODE models can be augmented to become partial differential equations (PDE) that allow for both temporal and spatial evolution. In particular, we'll be focusing on a class of PDEs known as *reaction-diffusion equations* where the motion of the species from one location to another is modelled by diffusion. We'll examine cases where these equations emit travelling wave solutions, as well as stationary patterns, and discuss how these arise in biological models.

1.1 Reaction-diffusion equations

Let's first derive the general form of a reaction-diffusion equation for single species. Specifically, we'll determine a PDE for function $u(\mathbf{x}, t)$, which depends on both space, $\mathbf{x}$, and time, t, and can represent, for example, the concentration of a particular molecule at point $\mathbf{x}$ and time t, or the population size of a particular organism at point $\mathbf{x}$ and time t. We'll develop the equation assuming $\mathbf{x} \in \mathbb{R}^3$, and reduce it later to cases of $x \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^2$.

Consider an arbitrary region in space, $\Omega \subset \mathbb{R}^3$, which has boundary $\partial\Omega$ and outward-facing unit-normal $\widehat{\mathbf{n}}(\mathbf{x})$. The total amount of chemical, or total number of organisms in Ω at time t is given by

$$\int_\Omega u(\mathbf{x}, t) d^3\mathbf{x}. \tag{1}$$

This quantity can change with time in one of two ways. The first way is by u evolving within Ω, which we will say is described by a function $f(u, \mathbf{x}, t)$. For example if $u(\mathbf{x}, t)$ represents a population size, $f(u, \mathbf{x}, t)$ could provide the rate of change through birth or death of members. Thus, $f(u, \mathbf{x}, t)$ is identical to the kinds of terms we were considering in our ODE models to describe the rates at which quantities change.

The second way (1) may change is by $u(\mathbf{x}, t)$ entering or leaving Ω through its boundary, $\partial\Omega$. If the motion of $u(\mathbf{x}, t)$ in space is given by the vector field $\mathbf{J}(\mathbf{x}, t)$, then the flux into Ω is $-\mathbf{J}(\mathbf{x}, t) \cdot \widehat{\mathbf{n}}(\mathbf{x})$ for $\mathbf{x} \in \partial\Omega$.

Putting these effects together, we have then that

$$\frac{d}{dt}\int_\Omega u(\mathbf{x}, t)d^3\mathbf{x} = \int_\Omega f(u, \mathbf{x}, t)d^3\mathbf{x} - \int_{\partial\Omega} \mathbf{J}(\mathbf{x}, t) \cdot \widehat{\mathbf{n}}(\mathbf{x})dS$$

Since the domain Ω does not change in time, we can move the time-derivative into the integral to obtain,

$$\frac{d}{dt}\int_\Omega u d^3\mathbf{x} = \int_\Omega \frac{\partial u}{\partial t} d^3\mathbf{x}.$$

We also have changed from a total derivative to a partial derivative since the points $\mathbf{x}$ do not change in time (this does not apply in fluid dynamics or continuum mechanics!). Additionally, we use the divergence theorem to rewrite the surface integral involving the flux as

$$\int_{\partial\Omega} \mathbf{J} \cdot \widehat{\mathbf{n}} dS = \int_\Omega \boldsymbol{\nabla} \cdot \mathbf{J} d^3\mathbf{x}.$$

Putting everything together yields

$$\int_\Omega \frac{\partial u}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{J} - f d^3\mathbf{x} = 0.$$

The next step in our derivation is to recall that the domain Ω is arbitrary and thus, for this to hold for all possible Ω, the integrand must be zero and, therefore,

$$\frac{\partial u}{\partial t} = f - \boldsymbol{\nabla} \cdot \mathbf{J}.$$

The last thing to do is specify $\mathbf{J}$. Here, we'll assume that the tendency is for u to move from regions of higher values to lower values in the direction opposite its gradient, such that $\mathbf{J} = -D\boldsymbol{\nabla} u$, where D is a positive constant called the diffusion coefficient. This relationship is known as Fick's law and describes the spread of chemical concentrations due to Brownian motion of their constituent molecules, as well as the effective motion over time of populations of organims that randomly change their direction of motion. With this expression for $\mathbf{J}$, we arrive at the final general form,

$$\frac{\partial u}{\partial t} = f + D\nabla^2 u.$$

1.2 Fisher-Kolmogorov (FK) Equation

The first model we will examine is attributed to Ronald Fisher for his work published 1937 that models the spread of an advantageous gene mutuation in a population. We'll get to Kolmogorov's contribution shortly.

The FK equation provides a natural entry point into our discussion of reaction-diffusion equations and spatial dynamics since in this equation

$$f(u) = ku\,(1-u)\,,$$

which corresponds to logistic growth with unit carrying capacity. In the context of Fisher's model, u is the percentage of the population carrying the advantageous gene. We'll also consider the case of one spatial dimension such that $\nabla^2 \to \partial^2/\partial x^2$, yielding

$$\frac{\partial u}{\partial t} = ku\,(1-u) + D\frac{\partial^2 u}{\partial x^2}.$$

To nondimensionalise, we write $t = \tau/k$, and $x = (D/k)^{1/2}X$ and upon substituting these expressions into the FK equation, we find

$$\frac{\partial u}{\partial \tau} = u\,(1-u) + \frac{\partial^2 u}{\partial X^2}. \tag{2}$$

1.2.1 Travelling waves

Recall from our discussion of the logistic equation that the ODE

$$\frac{du}{d\tau} = u\,(1-u) \tag{3}$$

has two fixed points: $u = 0$, which is unstable, and $u = 1$, which is stable. In the context of the reaction-diffusion PDE, these constant values of u are *homogeneous steady solutions* of the spatially extended system since they also satisfy

$$u\,(1-u) + \frac{\partial^2 u}{\partial X^2} = 0.$$

Let's now consider the FK equation with the conditions that

$$u \to 1 \text{ as } X \to -\infty \text{ and } u \to 0 \text{ as } X \to \infty, \tag{4}$$

which, therefore, requires the solution to transition from $u = 1$ to $u = 0$ over some region of space. Knowing what we know about the stabilities of $u = 1$ and $u = 0$, we expect that the transition region will not be stationary and move to the right, allowing $u = 1$ region to expand. Based on this intuition, we'll search for a *travelling wave* solution, which has the form

$$u(X, \tau) = U(\xi), \ \xi = X - c\tau$$

where the constant $c \geq 0$ is the wave speed, which is *unknown* and must be determined as part of the problem. The solution, therefore, can be viewed as having a fixed shape that is shifted to the right by the amount $c\tau$ at time τ.

With this ansatz, we'll have

$$\frac{\partial u}{\partial \tau} = -c\frac{dU}{d\xi},$$
$$\frac{\partial^2 u}{\partial X^2} = \frac{d^2 U}{d\xi^2},$$

which turns the FK equation into the second-order ODE

$$\frac{d^2 U}{d\xi^2} = -U(1 - U) - c\frac{dU}{d\xi}.$$

Introducing variable $V = dU/d\xi$, we arrive at the system of first-order ODEs

$$\frac{dU}{d\xi} = V, \tag{5}$$
$$\frac{dV}{d\xi} = -U(1 - U) - cV. \tag{6}$$

We're now in a position to apply all of the tools we are familiar with for analysing systems of ODEs. Before doing so, let's pause to recall some important facts about the solution. The solution will be a function of ξ. Based on conditions (4), as ξ increases, the solution will go from $U = 1$ to $U = 0$. Next, recall that U represents a population size. Thus, any solutions involving $U < 0$ are not realistic and should be discarded. Finally, since $V = dU/d\xi$, if U decreases monotonically, then we must have $V < 0$ for all ξ.

Let's now proceed by performing a linear stability analysis about the fixed points of the system, which are $(0, 0)$ and $(1, 0)$. The Jacobian is given by

$$J(U,V) = \left[\begin{array}{cc} 0 & 1 \\ 2U-1 & -c \end{array} \right],$$

which, when evaluated at the fixed points yield

$$J(0,0) = \left[\begin{array}{cc} 0 & 1 \\ -1 & -c \end{array} \right], \text{ and } J(1,0) = \left[\begin{array}{cc} 0 & 1 \\ 1 & -c \end{array} \right].$$

The eigenvalues of the Jacobian evaluated at $(0,0)$ are

$$\lambda_{\pm} = \frac{1}{2}\left(-c \pm \sqrt{c^2-4}\right),$$

and we see that it will be a stable node if $c > 2$, or a stable spiral if $c < 2$. For the fixed point $(1,0)$, the eigenvalues of the Jacobian are

$$\lambda_{\pm} = \frac{1}{2}\left(-c \pm \sqrt{c^2+4}\right),$$

making $(1,0)$ an saddle.

Let's step back again and scrutinise these results. First, if $c < 2$, the trajectory in phase space will spiral about the origin and for some values of ξ we will have $U < 0$. Recalling our discussion above, these solutions are unrealistic in the sense that they do not describe what can occur in the real system. Therefore, we'll only interest ourselves in the case where $c \geq 2$.

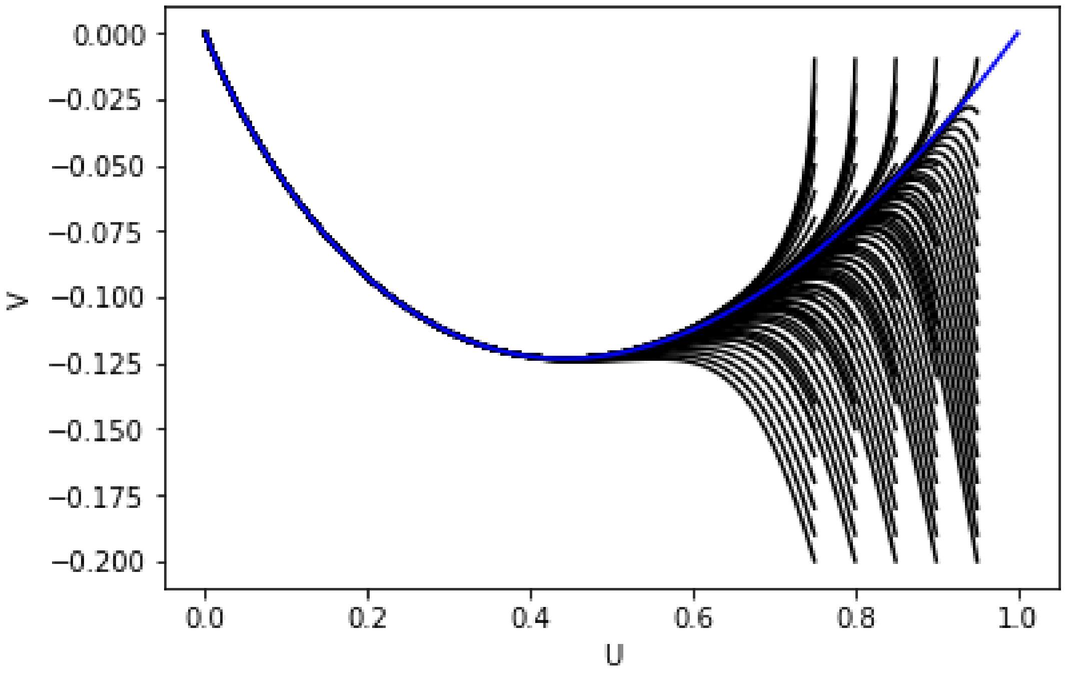
Given the stabilities of $u = 0$ and $u = 1$ for the ODE (3), the resulting stabilities of $(0,0)$ and $(1,0)$ for (6) may seem off. We again recall our discussion from above, namely, that we are interested in a solution that satisfies (4). Thus, we need the solution to move away from $(1,0)$ as ξ increases. Hence, we need $(1,0)$ to be unstable and near $(1,0)$, the solution will lie on the unstable manifold of the saddle. Further, the condition (4) means that the solution we seek 'starts' at the saddle $(1,0)$ and moves along the a trajectory that connects to the node at the origin. In the language of dynamical systems, this kind of trajectory is known as a heteroclinic orbit, or heteroclinic connection.

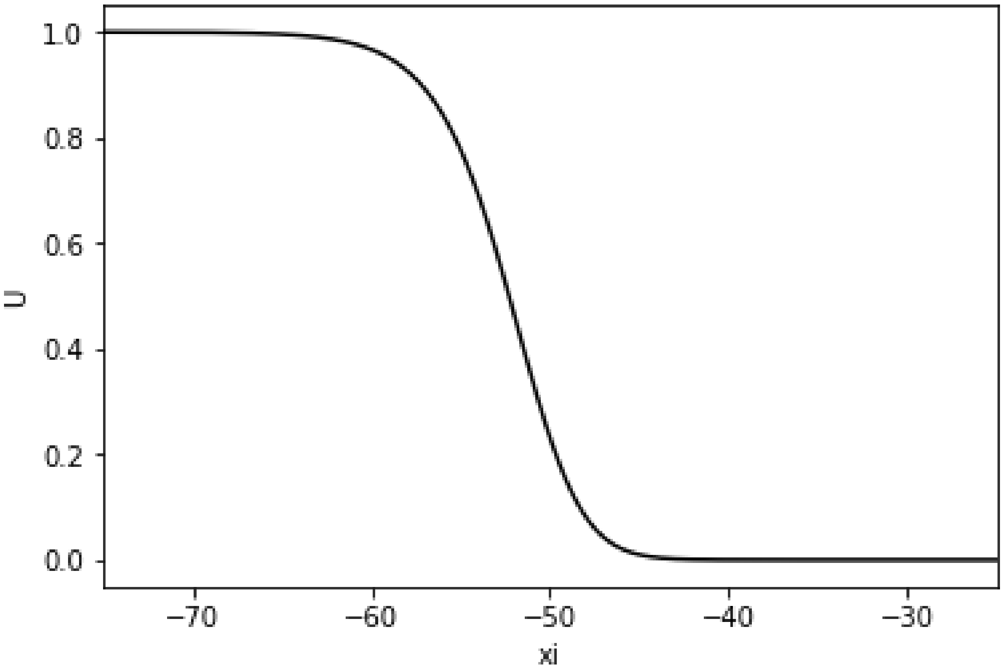
The Python cell below solves the system of ODEs numerically. The first figure shows in blue the trajectory connecting $(1,0)$ and $(0,0)$. Additionally, we observe that other trajectories rapidly approach the heteroclinic orbit that connects $(1,0)$ and $(0,0)$. The second figure shows $U(\xi)$, which will be the shape of the travelling wave for the specified value of c.

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg
```

```
c = 2
eps = 1e-9

def du_dt(usol, xi):
    return [usol[1], -usol[0]*(1 - usol[0]) - c*usol[1]]

J = np.array([[0, 1], [1, -c]])
lam, evec = linalg.eig(J)

xi = np.linspace(-100, 100, 10000)
plt.figure(0)
icsu = np.arange(0.75, 1, 0.05)
icsv = np.arange(-0.2, 0.0, 0.01)
for i in icsu:
    for j in icsv:
        u0 = [i, j]
        us = odeint(du_dt, u0, xi)
        U = us[:,0]
        V = us[:,1]
        plt.plot(U,V,'k-');

u0 = [1 - eps*evec[0,0], 0 - eps*evec[1,0]]
us = odeint(du_dt, u0, xi)

vvec2 = us[:,0]
wvec2 = us[:,1]

plt.plot(vvec2,wvec2,'b-')
plt.xlabel('U')
plt.ylabel('V')

plt.figure(1)
plt.plot(xi,vvec2,'k-');
plt.xlabel('xi')
plt.ylabel('U')
plt.xlim(-75,-25);
```


0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−0.150
−0.175
−0.200
V
0.0
0.2
0.4
0.6
0.8
1.0
U


1.0
0.8
0.6
0.4
0.2
0.0
U
−70
−60
−50
−40
−30
xi

1.2.2 Wave speed

While the analysis above reveals that the Fisher-Kolmogorov equation can support travelling wave solutions with admissible wave speeds of $c \geq 2$, it does not give any information regarding the initial conditions, $u(X, 0)$, for which travelling waves emerge, and further, the specific values of c that these waves might have. This is where Kolmogorov's contribution enters. Going back to (2), he proved that if the initial condition satisfies,

$$u(X, 0) = u_0(X) \geq 0, \;\; u_0(X) = \begin{cases} 1 & \text{if } X \leq X_1 \\ 0 & \text{if } X \geq X_2 \end{cases} \tag{7}$$

with $X_1 < X_2$ and $u_0(X)$ is a continuous function of X in the interval $X_1 < X < X_2$, then the solution will eventually evolve into the travelling wave solution $U(\xi)$ with $c = 2$.

To better understand why this is the case, consider an intial condition with the property that

$$u(X, 0) \sim A\mathrm{e}^{-aX} \text{ as } X \to \infty.$$

Thus, when X is large, we will have $u^2 \ll 1$ and this term in the FK equation can be neglected. The solution to leading order will then obey the linearised FK equation

$$\frac{\partial u}{\partial \tau} = u + \frac{\partial^2 u}{\partial X^2}.$$

Assuming the travelling wave solution $u(X, \tau) = A\mathrm{e}^{-a(X - c\tau)}$, the linearised FK equations yields

$$c = \frac{1}{a} + a,$$

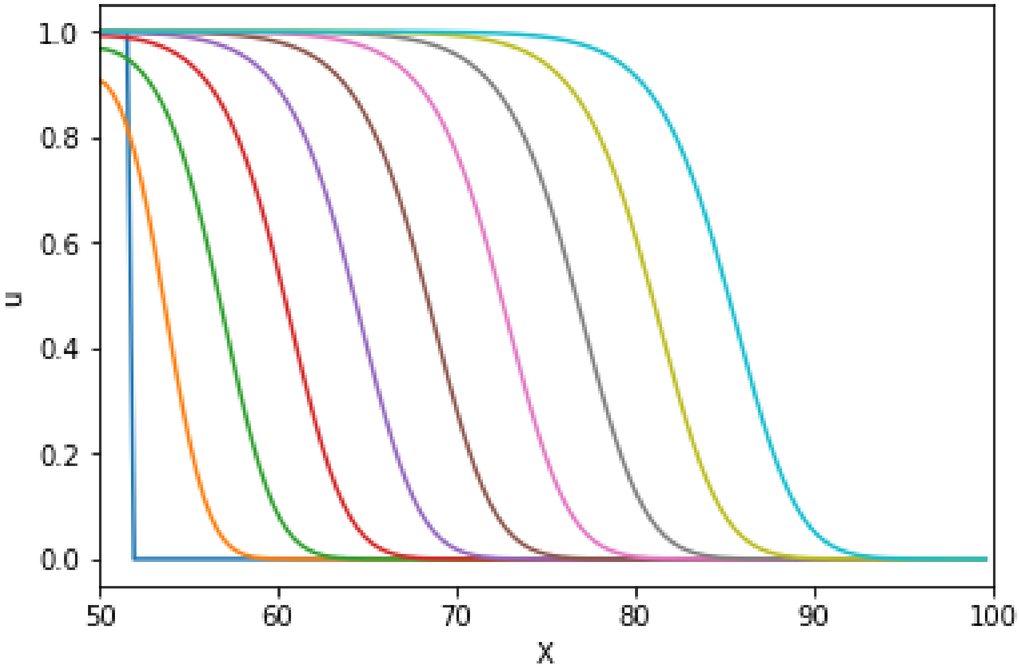
which gives $c \geq 2$ with $c = 2$ occurring when $a = 1$.

Now, we must use a key fact about the FK equation that we simply state: the FK equation (and its linearised version) is monotonic in its initial condition, meaning that if we consider two initial conditions with $u_1(X, 0) \leq u_2(X, 0)$, then $u_1(X, \tau) \leq u_2(X, \tau)$ for $\tau \geq 0$. Thus, if our initial condtion satisfies, $u(X, 0) \leq A\mathrm{e}^{-X}$, then $u(X - c\tau) \leq A\mathrm{e}^{-a(X-2\tau)}$. Therefore, for this inequality to hold, we would need $c \leq 2$. Since in general, however, $c \geq 2$, we see then that $c = 2$.

The Python cell below generates a numerical solution to the FK equation using a second-order finite-difference scheme to discretise the spatial coordinate with periodic boudary conditions applied at the ends of the domains. The initial condition is set to be a step function, which satisfies the condition (7). In the figure, the solution is plotted every $\Delta\tau = 2$, and since $c = 2$ there is a spacing of $\Delta X = 4$ between each curve. Additionally, we see that the shape of the solution closely resembles that given by the ODE system (6).

[2]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

N = 256
L = 100
dx = L/N
x = np.linspace(0, L, N, endpoint=False)
u0 = 0*x;
u0[int(N/2)-5:int(N/2)+5] = 1

D2 = np.eye(N, k = -1) - 2*np.eye(N) + np.eye(N,k = 1)
D2[N-1][0] = 1;
D2[0][N-1] = 1;
D2 = D2/dx**2

def du_dt(u, t):
    return u*(1-u) + D2.dot(u)

Nt = 10000
ts = np.linspace(0, 20, Nt)
us = odeint(du_dt, u0, ts)

timeind = np.linspace(0, Nt-1, int(Nt/1000), dtype=int)
for i in timeind:
        plt.plot(x, us[i, :]);

plt.xlim(50,100);
plt.xlabel('X')
plt.ylabel('u');
```



2 Spatial dynamics of a predator-prey system

With the FK equation, we saw how a reaction-diffusion equation can describe the spatial dynamics of an isolated population. Here, we will examine how reaction-diffusion equations can be coupled to model the spatial dynamics of interacting populations or chemical species. As we had done with our ODE models, interactions are included by considering a system of PDEs, which can be expressed as

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{f} + \mathcal{D}\nabla^2 \mathbf{u}.$$

where $\mathbf{u}(\mathbf{x}, t)$ and $\mathbf{f}(\mathbf{u}, \mathbf{x}, t)$ are vectors and now $\mathcal{D}$ is the diffusion matrix, which will assume is diagonal, and whose non-zero entries correspond to the diffusion coefficients of each species such that $\mathcal{D}_{ii} = D_i$. We see then, as we had done in the single species case, spatial dynamics are incorporated by adding diffusion to the ODE model of the system.

Here, we'll consider the system,

$$\begin{aligned}\frac{\partial N}{\partial t} &= rN\left(1 - \frac{N}{K}\right) - aNP + D_1\frac{\partial^2 N}{\partial x^2}\\ \frac{\partial P}{\partial t} &= bNP - cP + D_2\frac{\partial^2 P}{\partial x^2}\end{aligned}$$

where N is the prey population size, and P is the predator population size. In the absence of predators, the prey population obeys the FK equation with diffusion coefficient D_1, growth rate per capita, r, and carrying capacity, K. With predators, the prey are consumed at rate $-aNP$. The predator population grows in size proportional to its rate of prey consumption, bNP, dies off at a rate $-cP$, and spreads with a diffusion coefficient D_2.

We nondimensionalise the system by introducing

$$N = Ku, \quad P = \frac{r}{a}v, \quad t = \tau/r, \text{and}, x = (D_2/r)^{1/2}X$$

to obtain

$$\begin{aligned}\frac{\partial u}{\partial \tau} &= u\left(1 - u - v\right) + D\frac{\partial^2 u}{\partial X^2}\\ \frac{\partial v}{\partial \tau} &= \alpha v(u - \beta) + \frac{\partial^2 v}{\partial X^2}\end{aligned}$$

where $\alpha = bK/r$, $\beta = c/(bK)$, and $D = D_1/D_2$. Moving forward, we'll consider $D = 0$ and, therefore have

$$\frac{\partial u}{\partial \tau} = u\left(1 - u - v\right) \tag{8}$$

$$\frac{\partial v}{\partial \tau} = \alpha v(u - \beta) + \frac{\partial^2 v}{\partial X^2}. \tag{9}$$

This choice, which not only simplifies the analysis, corresponds to a situation where the prey is not able to move. This would apply, for example, to non-motile microorganisms that can be consumed by a motile predator, such as nematodes or another simple animal.

As we had done with the FK equation, let's remove the diffusive term and consider the system of ODEs

$$\begin{aligned}\frac{du}{d\tau} &= u\left(1 - u - v\right)\\ \frac{dv}{d\tau} &= \alpha v(u - \beta).\end{aligned}$$

This system has three fixed points, $(0, 0)$ and $(1, 0)$, where the predator population is zero, and $(\beta, 1 - \beta)$, where there is coexistence of the two populations. Note that we must require $\beta < 1$, to ensure $v > 0$ and we will take this to be the case in the following discussion. A linear stability analysis (left as an exercise) reveals that $(0, 0)$ and $(1, 0)$ are unstable, while $(\beta, 1 - \beta)$ is a stable node when $4\alpha < \beta/(1 - \beta)$, and a stable spiral for $4\alpha > \beta/(1 - \beta)$.

Returning to the PDE system, we'll proceed as we had done with the FK equation and search for travelling wave solutions of the form,

$$u(X,\tau) = U(\xi), \quad v(X,\tau) = V(\xi),$$

with, $\xi = X - c\tau$. Based on the fixed points of the ODE system, we will impose the conditions that

$$U \to 1 \text{ and } V \to 0, \text{ as } \xi \to \infty, \; U \to \beta \text{ and } V \to 1-\beta, \text{ as } \xi \to -\infty. \tag{10}$$

Substituting the travelling wave ansatz into (9), we obtain

$$\begin{aligned} -c\frac{dU}{d\xi} &= U(1-U-V) \\ -c\frac{dV}{d\xi} &= \alpha V(U-\beta) + \frac{d^2V}{d\xi^2}. \end{aligned}$$

Introducing $W = dV/d\xi$ and rearranging, we arrive at the three-dimensional system

$$\begin{aligned} \frac{dU}{d\xi} &= \frac{U}{c}(U+V-1) \\ \frac{dV}{d\xi} &= W \\ \frac{dW}{d\xi} &= \alpha V(\beta - U) - cW. \end{aligned}$$

Though the system is three-dimensional, we can analyse it much in the same way as we had done with the FK equation and establish the permissible range of wave speeds, c.

The system has fixed points $(0,0,0)$, $(1,0,0)$, and $(\beta, 1-\beta, 0)$. To obtain the trajectory that corresponds to the solution of the PDE that satifies conditions at infinity given by (10), we'll consider $(1,0,0)$ and $(\beta, 1-\beta, 0)$. To examine the stability of the fixed points, we compute the Jacobian,

$$J(U,V,W) = \begin{bmatrix} (2U+V-1)/c & U/c & 0 \\ 0 & 0 & 1 \\ -\alpha V & \alpha(\beta-U) & -c \end{bmatrix},$$

which, when evaluated at $(1,0,0)$ gives

$$J(1,0,0) = \begin{bmatrix} 1/c & 1/c & 0 \\ 0 & 0 & 1 \\ 0 & \alpha(\beta-1) & -c \end{bmatrix}.$$

This matrix has characteristic equation $(\lambda - 1/c)\,(\lambda(c+\lambda) - \alpha(\beta-1)) = 0$, and thus, the eigenvalues are

$$\lambda = 1/c,\ \ \lambda_{1,2} = \frac{1}{2}\left(-c \pm \sqrt{c^2 + 4\alpha(\beta-1)}\right).$$

From the expression for $\lambda_{1,2}$, we see that the solution will be oscillatory if $c^2 < 4\alpha(1-\beta)$. Since V cannot be less than zero, these oscillatory solutions can be discarded and therefore, the permissible wave speeds satisfy

$$c^2 \geq 4\alpha(1-\beta).$$

For fixed point $(\beta, 1-\beta, 0)$, the Jacobian is

$$J(\beta, 1-\beta, 0) = \begin{bmatrix} \beta/c & \beta/c & 0 \\ 0 & 0 & 1 \\ \alpha(\beta-1) & 0 & -c \end{bmatrix},$$

which has characteristic equation

$$\lambda^3 + \lambda^2\left(c - \frac{\beta}{c}\right) - \beta\lambda - \frac{\beta}{c}\alpha(\beta-1) = 0.$$

Unfortunately, this cubic polynomial is not as easily handled as the one encountered above. To better understand the solutions that arise in this case, we first compute the values of λ where the local maximum and minimum of the characteristic polynomial realised. These are given by,

$$\lambda_{\max}, \lambda_{\min} = -\frac{1}{3}\left(c - \frac{\beta}{c} \pm \sqrt{\left(c - \frac{\beta}{c}\right)^2 + 3\beta},\right)$$

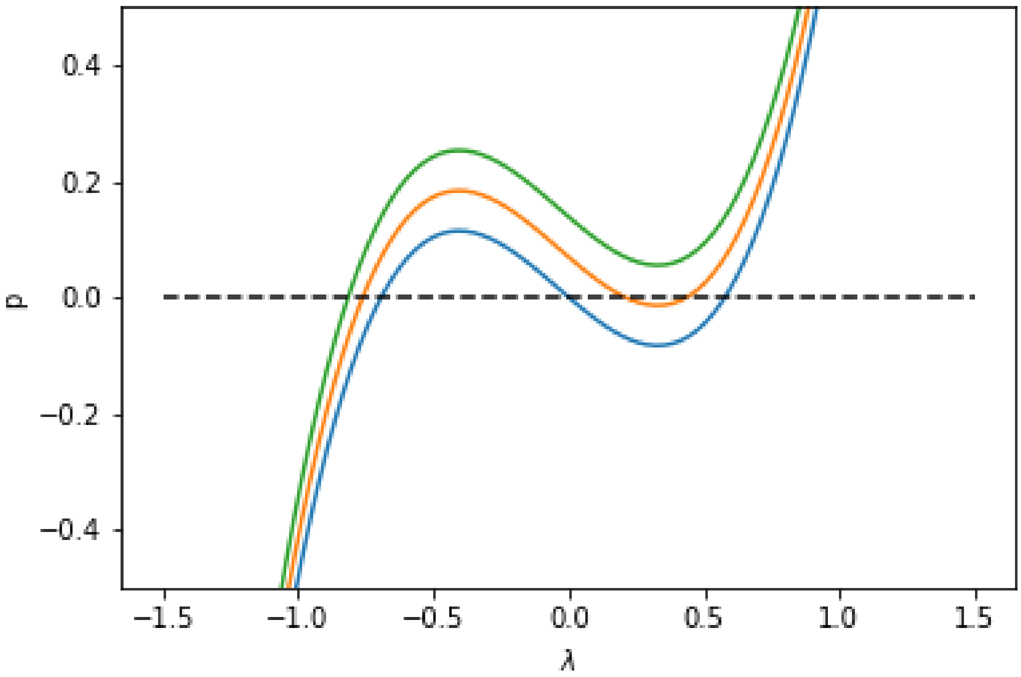
and we see that they do not depend on the parameter α. Note that these values are NOT necessarily eigenvalues, but rather the values where the characteristic polynomial attains its local maximum or minimum. Investigating further by taking $\alpha = 0$ (thus, v obeys the diffusion equation), we compute the eigenvalues to find

$$\lambda = 0, \quad \lambda = -c, \lambda = \beta/c$$

which are all real. Increasing α amounts to shifting the cubic upwards. Therefore, we see that as α increases from zero, all eigenvalues will remain real until α reaches a critical value, α_c, at which point, the characteristic equation emits $\lambda_{\min}$ as a repeated eigenvalue. For $\alpha > \alpha_c$, the local minimum lies above the x-axis and as a result there will be one real eigenvalue and two complex eigenvalues. This shown in the Python cell below that plots the cubic for different values of α.

[3]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

alpha1 = 0.0
alpha2 = 0.2
alpha3 = 0.4

beta = 0.4
c = (4*alpha2*(1 - beta))**0.5;

lam = np.linspace(-1.5,1.5,100);
p1 = lam**3 + lam**2*(c - beta/c) - beta*lam - beta*alpha1*(beta - 1)/c;
p2 = lam**3 + lam**2*(c - beta/c) - beta*lam - beta*alpha2*(beta - 1)/c;
p3 = lam**3 + lam**2*(c - beta/c) - beta*lam - beta*alpha3*(beta - 1)/c;

plt.plot(lam, p1)
plt.plot(lam, p2)
plt.plot(lam, p3)
plt.plot(lam,0*lam,'k--')
plt.ylabel('p')
plt.xlabel('$\lambda$');
plt.ylim(-0.5,0.5);
```



Thus, if $\alpha \leq \alpha_c$ and the unstable eigenvalues are real, we expect $(U(\xi), V(\xi))$ to vary monotonically with ξ, going from $(U, V) = (\beta, 1-\beta)$ to $(1, 0)$ to as ξ increases. Since the unstable eigenvalues are complex when $\alpha > \alpha_c$, the solution will be oscillate as ξ increases. Previously, we discarded these solutions on the grounds that they would yield negative values of a population size. In those instances, however, we were dealing with fixed points where one population size was zero. Here, we have $(\beta, 1-\beta)$, so the solution may safely oscillate without ever becoming negative. Thus, we can permit oscillatory wavefronts. While this is a fundamental difference with the FK equation where oscillations were not possible, like the FK equation the wave speed is given by the minimum possible value, namely,

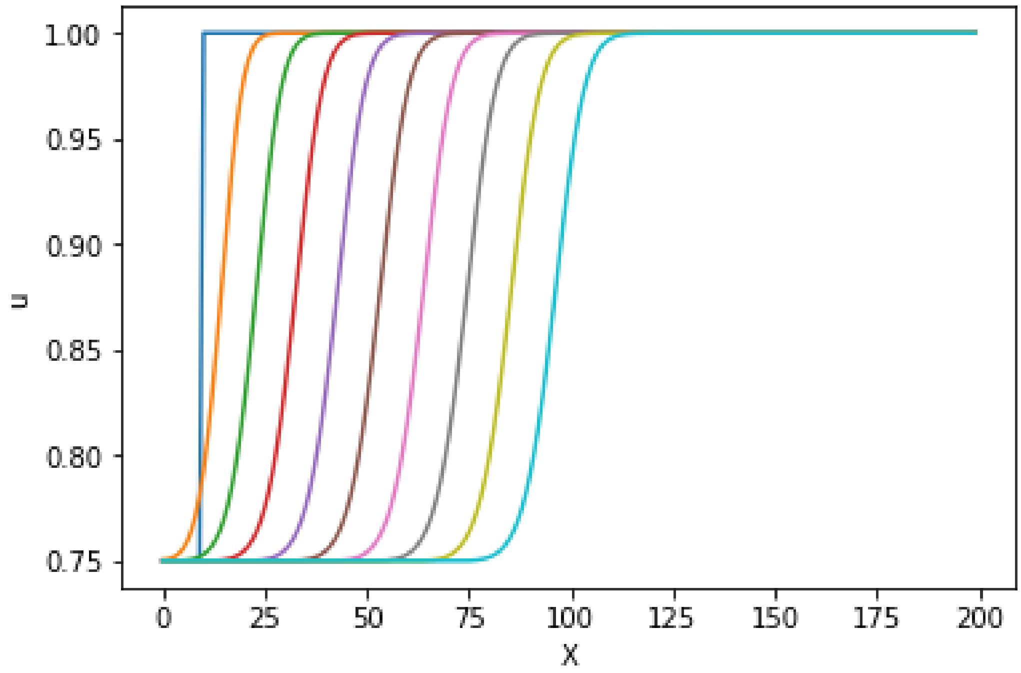
$$c^2 = 4\alpha(1-\beta).$$

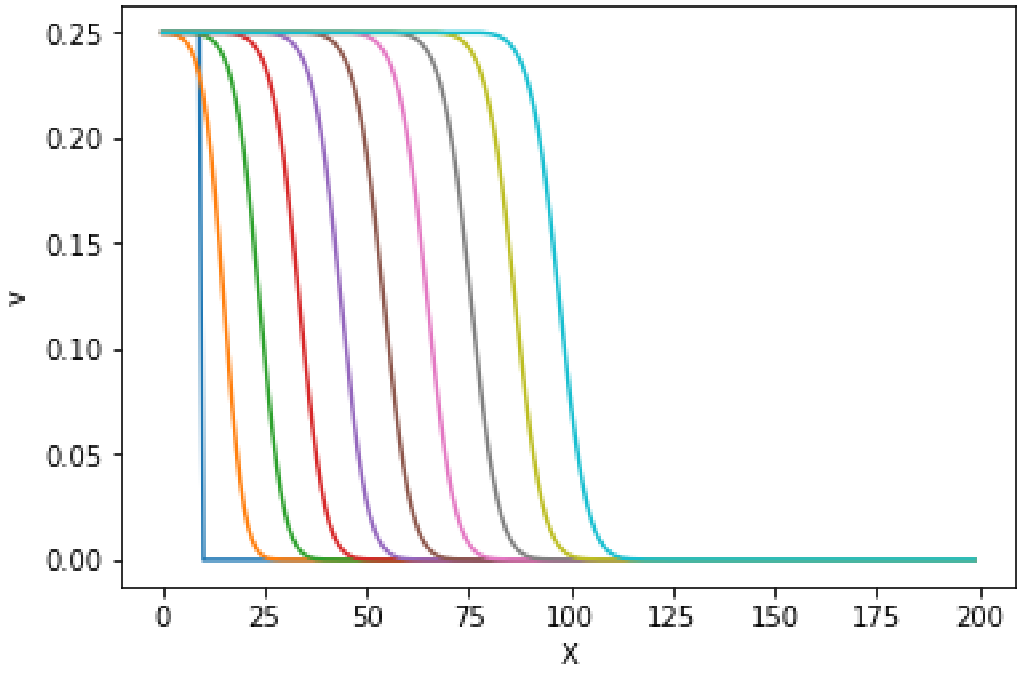
Both the oscillating wavefront, and the speed of the wave can be observed in the numerical solution generated below.

[1]:

```
## CODE FINDING A NUMERICAL SOLUTION TO THE PDE SYSTEM

import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

N = 200
L = 200
dx = L/N

alpha = 1;
beta = 0.75;

x = np.linspace(0, L, N, endpoint=False)
u0 = 0*x + 1;
u0[0:10] = beta
#u0[11:20] = 1

v0 = 0*x;
v0[0:10] = 1 - beta;
#v0[11:20] = 0;

D2 = np.eye(N, k = -1) - 2*np.eye(N) + np.eye(N,k = 1)
#D2[0][N-1] = 1;
#D2[N-1][0] = 1;
D2 = D2/dx**2

D2[0][0] = -1;
```

```
D2[0][1] = 1;
D2[N-1][N-1] = 1;
D2[N-1][N-2] = -1;


usol0 = np.concatenate((u0, v0))

def du_dt(usol, t):
    u = usol[0:N]
    v = usol[N:2*N]
    u1 = u*(1-u-v) + D2.dot(u)
    v1 = alpha*v*(u-beta) + D2.dot(v)
    return np.concatenate((u1, v1))

Nt = 10000
ts = np.linspace(0, 100, Nt)
us = odeint(du_dt, usol0, ts)

timeind = np.linspace(0, Nt-1, int(Nt/1000), dtype=int)
for i in timeind:
        plt.figure(0)
        plt.plot(x,us[i, 0:N]);
        plt.figure(1)
        plt.plot(x,us[i, N:2*N]);

plt.figure(0)
plt.xlabel('X')
plt.ylabel('u');

plt.figure(1)
plt.xlabel('X')
plt.ylabel('v');
```


1.00
0.95
0.90
0.85
0.80
0.75
u
0
25
50
75
100
125
150
175
200
X


0.25
0.20
0.15
0.10
0.05
0.00
v
0
25
50
75
100
125
150
175
200
X

```
[ ]:
```

Mathematical Biology - Week 8

November 16, 2022

0.1 Spatial spread of epidemics

Spatial dynamics can play a role in epidemics as disease may spread by infected individuals moving from one location to another. Let's consider again the SIS model introduced in our discussion of one dimensional systems, and extend it to include spatial dynamics such that we have,

$$\frac{\partial S}{\partial t} = -rSI + aI + D\frac{\partial^2 S}{\partial x^2} \tag{1}$$

$$\frac{\partial I}{\partial t} = rSI - aI + D\frac{\partial^2 I}{\partial x^2}, \tag{2}$$

where the susceptibles and the infectives have the same diffusion coefficient. We observed previously with the ODE system that since $N = S + I$ is constant, the SIS model reduces to the logistic equation. Here, we can show a similar simplification. If we add the two equations appearing in (2) and introduce $N = S + I$, we will have

$$\frac{\partial N}{\partial t} = D\frac{\partial^2 N}{\partial x^2},$$

which is the classic diffusion equation. Further, if at $t = 0$, N is constant, i.e. does not depend on x, and for domains of fixed size, the boundary conditions are such that $\partial N/\partial x = 0$, then $\partial N/\partial t = 0$ and N remains a constant for all time. Thus, we may write $S(x,t) = N - I(x,t)$ and substituting this expression for S into the equation for I in (2), we have

$$\frac{\partial I}{\partial t} = rI(N - I) - aI + D\frac{\partial^2 I}{\partial x^2}.$$

Rearranging slightly, we can cast the equation in the form

$$\frac{\partial I}{\partial t} = kI\left(1 - \frac{I}{K}\right) + D\frac{\partial^2 I}{\partial x^2},$$

where $k = rN - a$ and $K = (rN - a)/r$, which, upon writing $u = I/K$, becomes the FK equation! We see then just as the ODE SIS model reduces to the logistic equation, the PDE SIS model reduces to the logistic equation's PDE counterpart, the FK model. We should be careful and remind ourselves that we needed to assume also that the initial total population is spatially homogeneous, and boundary conditions that ensure members do not leave or enter the domain, but these are perhaps natural assumptions.

Based on our analysis of the FK equation, if we have the conditions that

$$S \to N, I \to 0 \text{ as } x \to \infty \text{ and } S \to N - K, I \to K \text{ as } x \to -\infty,$$

we see then that the infective population will move as a travelling wave, expanding from the region where $I = K$ to the region where $I = 0$. The wave speed will be given by $2\sqrt{kD} = 2\sqrt{(rN - a)D}$. Thus, we see that the spatial spread of the disease can be slowed by reducing the diffusion coefficient, by increasing the recovery rate, a, or by reducing the transmission constant, r.

0.1.1 Spread of rabies in a fox population

Let's now consider the slightly different model,

$$\frac{\partial S}{\partial t} = -rSI \tag{3}$$

$$\frac{\partial I}{\partial t} = rSI - aI + D\frac{\partial^2 I}{\partial x^2}, \tag{4}$$

in which the susceptible population does not diffuse, and the members of the infective population do not recover, but instead die and are removed from the population. This is a simple model to describe the spread of rabies in a fox population (believe it or not!). The lack of diffusion in the susceptible population is somewhat realistic in the sense that foxes obey territorial boundaries and do not venture far from their territory. Foxes infected by rabies, however, move about and readily invade the territory established by other foxes.

Assuming that in the absence of the disease we have $S = S_0$, we nondimensionalise the system by introducing

$$t = \tau/(rS_0), x = (D/(rS_0))^{1/2}X, S = S_0 s, I = S_0 h$$

to obtain

$$\frac{\partial s}{\partial \tau} = -sh \tag{5}$$

$$\frac{\partial h}{\partial \tau} = sh - \alpha h + \frac{\partial^2 h}{\partial X^2}, \tag{6}$$

where $\alpha = a/(rS_0)$. If we ignore the diffusive term for a moment and consider the ODE system, we see that provided $h = 0$, the population sizes do not change. Thus, any point on positive s-axis is a fixed point. We can go ahead and compute the Jacobian,

$$J(s,h) = \begin{bmatrix} -h & -s \\ h & s-\alpha \end{bmatrix}.$$

Taking $h = 0$, we have

$$J(s,0) = \begin{bmatrix} 0 & -s \\ 0 & s-\alpha \end{bmatrix},$$

which has eigenvalues $\lambda = 0$ and $\lambda = s - \alpha$. If $s < \alpha$ then the fixed point at s will be stable. If, however, $s > \alpha$, then the fixed point will be unstable. We will see how this affects the spatial dynamics shortly, but for now, we expect any finite values of the infective population to be transient, and after a period of time, the infective population size will return to zero. However, since recovery is not possible, any transient infective population will have the effect of reducing the fox population after the disease has gone away.

With this information in mind, we'll now look for travelling wave solutions of the system (6). First, since the infective population returns to zero over time, we'll assume that it is spatially localised in the sense of

$$h \to 0 \text{ as } x \to \pm\infty.$$

As the infective population wave moves along, it will leave a reduced susceptible population in its wake. This new population size, though constant, will need to be determined as part of the problem. Upstream from the infective wave, we'll simply have the normal fox population size. Based on this, for the susceptible population we'll have the boundary conditions,

$$\frac{ds}{dx} \to 0 \text{ as } x \to -\infty \text{ and } s \to 1 \text{ as } x \to \infty.$$

Assuming the populations are travelling waves, we consider $s(x,t) = \Sigma(\xi)$ and $h(x,t) = H(\xi)$ with $\xi = x - ct$, where c is the wave speed. On substituting these expressions into (6), we find

$$-c\frac{d\Sigma}{d\xi} = -\Sigma H \tag{7}$$

$$-c\frac{dH}{d\xi} = \Sigma H - \alpha H + \frac{d^2 H}{d\xi^2}. \tag{8}$$

Introducing $P = dH/d\xi$ and rearranging, we arrive at the system

$$
\begin{aligned}
\frac{d\Sigma}{d\xi} &= \frac{\Sigma H}{c},\\
\frac{dH}{d\xi} &= P,\\
\frac{dP}{d\xi} &= -cP + \alpha H - \Sigma H.
\end{aligned}
$$

Recalling our analyses of the previous models, we identified travelling wave solutions as corresponding to trajectories in phase space that connect two fixed points of the system of ODEs that are obtained when one assumes the travelling wave solution. Here, the situation is a bit different. We know that one fixed point will be $(1,0,0)$, which the solution tends to as $\xi \to \infty$, but we don't know the other just yet. Estabilishing this fixed point will give the size of the population that remains after the infective wave has passed.

Let's focus first on the fixed point that we do know, $(1,0,0)$. To examine its stability, we compute the Jacobian,

$$J(\Sigma, H, P) = \begin{bmatrix} H/c & \Sigma/c & 0 \\ 0 & 0 & 1 \\ -H & \alpha - \Sigma & -c \end{bmatrix},$$

which, when evaluated at $(1,0,0)$ gives,

$$J(1,0,0) = \begin{bmatrix} 0 & 1/c & 0 \\ 0 & 0 & 1 \\ 0 & \alpha - 1 & -c \end{bmatrix}.$$

The eigenvalues of this matrix are

$$\lambda = 0, \;\; \lambda = \frac{1}{2}\left(-c \pm \sqrt{c^2 + 4(\alpha - 1)}\right).$$

Based on our considerations of the system of ODEs, we're interested in this fixed point corresponding to a case where $\alpha < 1$. If this is the case, in order to avoid oscillatory solutions that produce negative population sizes, the wave speed must satisfy,

$$c^2 \geq 4(1-\alpha).$$

This condition also means that the fixed point $(1,0,0)$ is stable, as it should be to allow the solution to approach $(1,0,0)$ as $\xi \to \infty$.

To determine the population size that remains after the passing of the wave, we revert back to the second-order system (8) and use the first equation to express H as

$$H = \frac{c}{\Sigma}\frac{d\Sigma}{d\xi}.$$

After substituting this expression for H into the second equation, we have

$$\frac{d^2H}{d\xi^2} + c\frac{dH}{d\xi} + \frac{c}{\Sigma}\frac{d\Sigma}{d\xi}(\Sigma - \alpha) = 0.$$

Integrating this equation with respect to ξ gives

$$\frac{dH}{d\xi} + cH + c\Sigma - c\alpha \log \Sigma = K,$$

where K is the constant of integration. We can determine K by imposing the conditions that $\Sigma = 1$ and $H = dH/d\xi = 0$ as $\xi \to \infty$. In doing so, we find that $K = c$.

Finally, we impose the condition that $H = dH/d\xi = 0$ as $\xi \to -\infty$, to obtain the expression

$$\sigma - \alpha \log \sigma = 1.$$

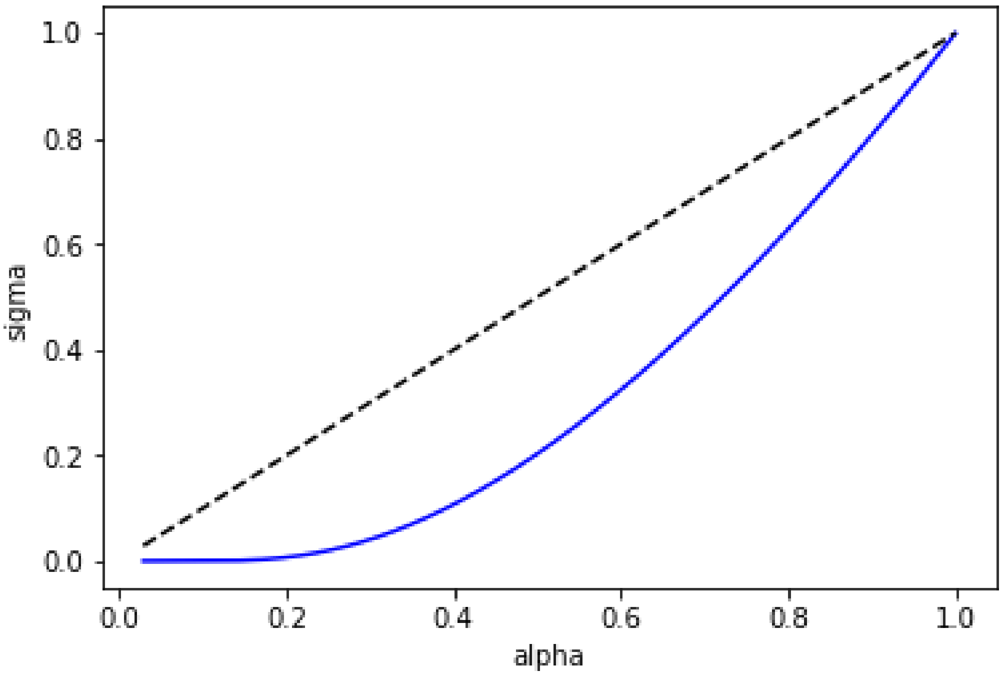
for σ, the value of Σ as $\xi \to -\infty$. Rearranging this expression ever so slightly, we have

$$\alpha = \frac{\sigma - 1}{\log \sigma}.$$

The Python cell below plots this function, giving the value of σ as a function of α.

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

sigma = np.linspace(1e-15, 0.999, 1000)
alpha = (sigma - 1)/(np.log(sigma))

plt.plot(alpha,sigma,'b-')
plt.plot(alpha,alpha,'k--');
plt.xlabel('alpha');
plt.ylabel('sigma');
```



We see that provided $\alpha < 1$, we have $\sigma < \alpha$, which, if you recall our discussion above, corresponds to stable fixed points of the ODE system. This analysis also reveals another way of interpreting $\alpha \geq 1$.

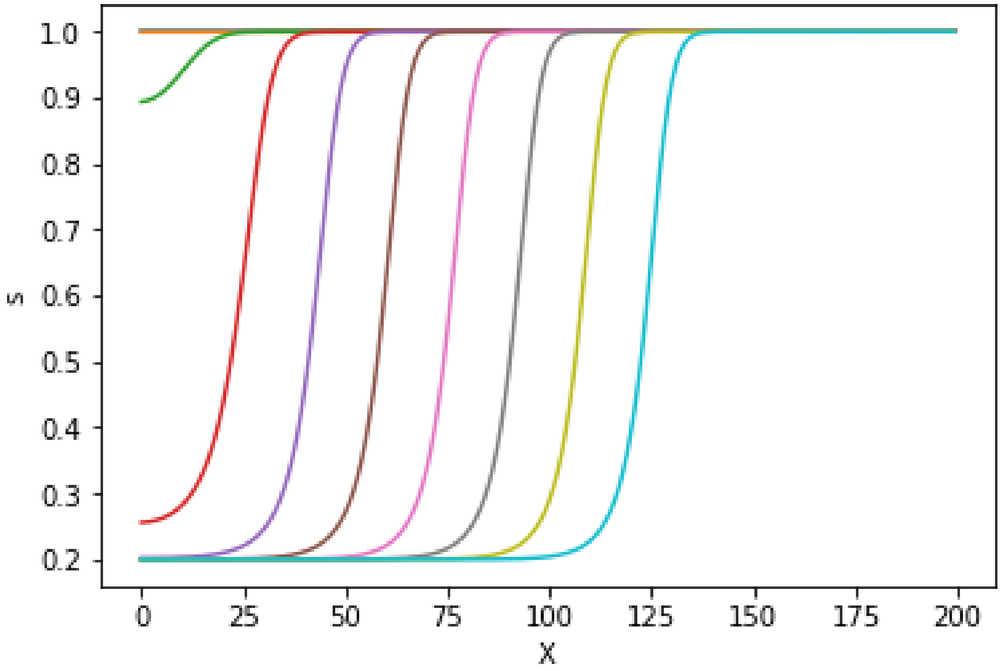
We see that for $\alpha = 1$, we will have $\sigma = 1$ and therefore, the uninfected populations will be $\Sigma = 1$ at each point in space. This means that there will be no wave propagating. In fact, this is the case for all $\alpha > 1$. Recall that $\alpha = a/(rS_0)$, which gives the ratio of the timescale to infect members to the timescale of death after infection. We see then that for $\alpha \geq 1$, the infected foxes will die before having a chance to infect others, eliminating the possibility of an epidemic.

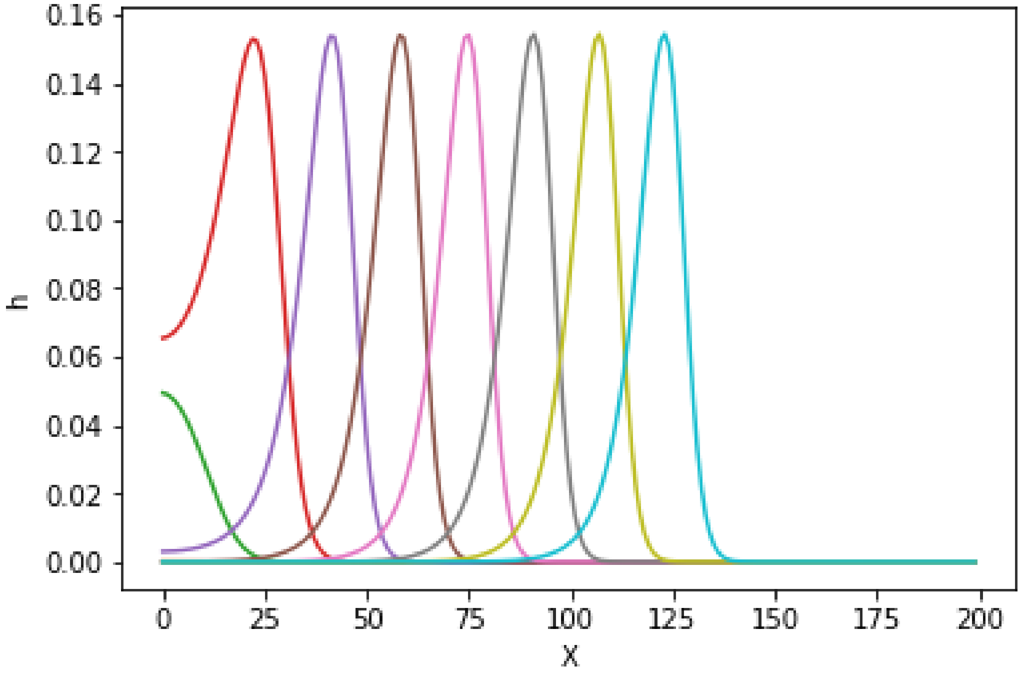
The numerical solutions generated using the Python cell below show the travelling waves for the susceptible and infective fox populations. Observe the localised nature of infective fox population that propagates to the right and leaves a reduced fox population in its wake.

[2]:

```
## CODE FINDING A NUMERICAL SOLUTION TO THE PDE SYSTEM
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

N = 200
L = 200
dx = L/N

sigma = 0.2;
```

```
alpha = (sigma - 1)/(np.log(sigma))

x = np.linspace(0, L, N, endpoint=False)

s0 = 0*x + 1;
#s0[0:10] = sigma;

h0 = 0*x;
h0[0:10] = 1e-6

D2 = np.eye(N, k = -1) - 2*np.eye(N) + np.eye(N,k = 1)
#D2[0][N-1] = 1;
#D2[N-1][0] = 1;
D2 = D2/dx**2

D2[0][0] = -1;
D2[0][1] = 1;
D2[N-1][N-1] = 1;
D2[N-1][N-2] = -1;


shsol0 = np.concatenate((s0, h0))

def du_dt(shsol, t):
    s = shsol[0:N]
    h = shsol[N:2*N]
    s1 = -s*h
    h1 = s*h - alpha*h + D2.dot(h)
    return np.concatenate((s1, h1))

Nt = 10000
ts = np.linspace(0, 100, Nt)
sh = odeint(du_dt, shsol0, ts)

timeind = np.linspace(0, Nt-1, int(Nt/1000), dtype=int)

for i in timeind:
        plt.figure(0)
        plt.plot(x,sh[i, 0:N]);
        plt.figure(1)
        plt.plot(x,sh[i, N:2*N]);

plt.figure(0)
plt.xlabel('X')
plt.ylabel('s');

plt.figure(1)
plt.xlabel('X')
plt.ylabel('h');
```





1 Pattern formation

Patterns can be readily observed throughout the biological world. While the stripes on a zebra stripes, or spots on a leopard may immediately jump to mind, perhap a more profound examples occur during embryo developement and allow organisms to develop along a stereotyped bodyplan, ensuring that vital organs are correctly situated. This aspect of development, known as morphogenesis, involves chemicals known as morphogens that react and spread to eventually reach a steady, but nontrivial, distribution of their concentrations – the pattern. The established concentration pattern can then be read and followed by cells, allowing them to divide or differentiate in a particular way.

In this part of the course, we'll explore how diffusion-driven instabilities of morphogen concentrations provide a mechanism by which spatial chemical patterns can emerge. The reaction-diffusion equation-based theory behind this mechanism was established in 1952 in a now classical paper by Alan Turing.

1.1 Turing instability

Consider the general system of reaction diffusion equations

$$\frac{\partial u}{\partial t} = f(u,v) + D_1 \nabla^2 u \tag{9}$$

$$\frac{\partial v}{\partial t} = g(u,v) + D_2 \nabla^2 v \tag{10}$$

for morphogen concentrations, u and v, in a region of space $\Omega \subset \mathbb{R}^3$. We'll assume that no morphogens enter Ω through its boundary $\partial\Omega$ and impose no-flux boundary conditions

$$\widehat{\mathbf{n}} \cdot \boldsymbol{\nabla} u = 0 \text{ and } \widehat{\mathbf{n}} \cdot \boldsymbol{\nabla} v = 0$$

for $\mathbf{x} \in \partial\Omega$. Thus, since morphogens cannot enter the domain, any pattern that results is purely do the reactions and diffusion of the chemical concentrations that are already present.

What we'd like to do is establish a set of conditions that must be met in order for (10) to emit patterns. To do so, we begin by considering the system of ODEs,

$$\frac{du}{dt} = f(u,v)$$

$$\frac{dv}{dt} = g(u,v).$$

In order for a pattern to emerge, this system must have a stable fixed point, $u = u^*$ and $v = v^*$ say. Therefore, the Jacobian

$$J = \left[\begin{array}{cc} \frac{\partial f}{\partial u}(u^*, v^*) & \frac{\partial f}{\partial v}(u^*, v^*) \\ \frac{\partial g}{\partial u}(u^*, v^*) & \frac{\partial g}{\partial v}(u^*, v^*) \end{array} \right] = \left[\begin{array}{cc} J_{11} & J_{12} \\ J_{21} & J_{22} \end{array} \right]$$

must satisfy $\operatorname{trace}(J) = J_{11} + J_{22} < 0$ and $\det(J) = J_{11}J_{22} - J_{12}J_{21} > 0$ to ensure that the fixed point is stable. These will be the first conditions that must be satisfied.

Returning now to the PDE system, we'll assess the stability of the homogeneous steady solutions, $u(\mathbf{x}, t) = u^*$ and $v(\mathbf{x}, t) = v^*$. Namely, we'll consider

$$u(\mathbf{x}, t) = u^* + \epsilon U(\mathbf{x}, t) \qquad (11)$$
$$v(\mathbf{x}, t) = v^* + \epsilon V(\mathbf{x}, t) \qquad (12)$$

where $0 < \epsilon \ll 1$ and determine whether the perturbations $U(\mathbf{x}, t)$ and $V(\mathbf{x}, t)$ grow with time, or decay away, allowing the system to return to the homogeneous steady value. You may be thinking that this is a pointless exercise since we're working from the assumption that u^* and v^* are stable solutions. This, however, this is only case for the ODE system, not the PDE. Thus, allowing U and V to depend on space opens up new avenues for u^* and v^* to be unstable, and in order for patterns to emerge, the homogeneous solution must be unstable.

Substituting (12) into (10), we have

$$\epsilon \frac{\partial U}{\partial t} = f(u^* + \epsilon U, v^* + \epsilon V) + D_1 \epsilon \nabla^2 U \qquad (13)$$
$$\epsilon \frac{\partial v}{\partial t} = g(u^* + \epsilon U, v^* + \epsilon V) + D_2 \epsilon \nabla^2 V. \qquad (14)$$

Since $\epsilon \ll 1$, we expand can f and g such that

$$\left[\begin{array}{c} f(u^* + \epsilon U, v^* + \epsilon V) \\ g(u^* + \epsilon U, v^* + \epsilon V) \end{array} \right] = \left[\begin{array}{c} f(u^*, v^*) \\ g(u^*, v^*) \end{array} \right] + \epsilon J \left[\begin{array}{c} U \\ V \end{array} \right] + O(\epsilon^2),$$

and consider the linear dynamics near the homogeneous steady solution. Writing $u^* + \epsilon U$ and $v^* + \epsilon V$ and using the expansion in (10) for f and g, we arrive at

$$\frac{\partial \mathbf{W}}{\partial t} = J\mathbf{W} + \mathcal{D}\nabla^2 \mathbf{W}, \qquad (15)$$

where $\mathbf{W} = [U, V]^T$ and $\mathcal{D}$ is the diagonal diffusion matrix which has non-zero entries $\mathcal{D}_{ii} = D_i$.

We can solve this linear equation using separation of variables where we assume that the solution takes the form

$$\mathbf{W}(\mathbf{x},t) = \boldsymbol{\zeta} F(t)\phi(\mathbf{x}),$$

where $\boldsymbol{\zeta}$ is a constant vector, $F(t)$ is only a function of time, and $\phi(\mathbf{x})$ is only a function of space. Substituting this expression into for $\mathbf{W}$ into (15) yields

$$\boldsymbol{\zeta}\phi\frac{dF}{dt} = F\phi J\boldsymbol{\zeta} + F\mathcal{D}\boldsymbol{\zeta}\nabla^2\phi. \tag{16}$$

Dividing through by F and ϕ gives

$$\frac{\boldsymbol{\zeta}}{F}\frac{dF}{dt} = J\boldsymbol{\zeta} + \frac{\mathcal{D}\boldsymbol{\zeta}}{\phi}\nabla^2\phi. \tag{17}$$

Since the term $J\boldsymbol{\zeta}$ is independent of both t and $\mathbf{x}$, in order for this equation to make sense, the other terms must be as well. Accordingly, we find that the function F is given by the eigenvalue problem

$$\frac{dF}{dt} = \lambda F,$$

and hence,

$$F = e^{\lambda t}, \tag{18}$$

while ϕ is given by another the eigenvalue problem,

$$\nabla^2\phi + k^2\phi = 0, \;\; \mathbf{x} \in \Omega, \tag{19}$$

$$\widehat{\mathbf{n}} \cdot \boldsymbol{\nabla}\phi = 0, \;\; \mathbf{x} \in \partial\Omega. \tag{20}$$

Note that constants λ and k^2 are the eigenvalues and must be determined as well. It is important to also note that in general there will be a countably infinite number of solutions to (20), each with their own value of k^2. The exact forms of the possible eigenfunction, $\phi(\mathbf{x})$ will depend on spatial dimension, as well as on the shape and size of the domain Ω. Based on the properties of the Laplacian, we express the eigenvalue as k^2, as it must be real and greater than or equal to zero. The specific choice of k^2 is also a natural one for rectangular domains as we will see later when we examine that case in detail.

Combining these expressions with (17) and rearranging, we arrive at a third and final eigenvalue problem

$$(J - k^2\mathcal{D})\boldsymbol{\zeta} = \lambda\boldsymbol{\zeta},$$

that provides the relationship between λ and k^2. With $F(t)$ given by (18), we can write the solution as

$$\mathbf{W} = \boldsymbol{\zeta}\phi(\mathbf{x})\mathrm{e}^{\lambda t}$$

where $\lambda(k^2)$ is determined from

$$\det(\lambda I - J + k^2\mathcal{D}) = 0. \tag{21}$$

We see, therefore, that if $\mathrm{Re}(\lambda) > 0$, perturbations with spatial dependence $\phi(\mathbf{x})$ will grow exponentially with time, leading to the growth and eventual establishment of the pattern. Perturbations that correspond to $\mathrm{Re}(\lambda) < 0$ will instead decay away.

1.1.1 Conditions for pattern formation

In order for pattern formation to occur, the homogeneous steady state must be unstable for some perturbations $\phi(\mathbf{x})$. Here, we will establish conditions that must be satisfied in order to have $\mathrm{Re}(\lambda) > 0$ for some range of k.

Based on (21), we see that in order for $\mathrm{Re}(\lambda) > 0$, the matrix $M = J - k^2\mathcal{D}$ must have,

$$\mathrm{trace}(M) = J_{11} + J_{22} - k^2(D_1 + D_2) > 0, \tag{22}$$

or

$$\det(M) = (J_{11} - k^2D_1)(J_{22} - k^2D_2) - J_{21}J_{12} < 0. \tag{23}$$

Let's first examine the condition on the trace, (22). First, we recall that $J_{11} + J_{22} = \mathrm{trace}(J) < 0$, since the fixed point of the system of ODEs is stable (see above). We also have that $-k^2\mathrm{trace}(D) < 0$, since $k^2 > 0$ and $D_1 > 0$, $D_2 > 0$. Thus, $\mathrm{trace}(M) < 0$ and the condition (22) cannot be satisfied.

The only way then that we can have $\mathrm{Re}(\lambda) > 0$ is if (23) is satified. We can, therefore, use this condition to determine if $\mathrm{Re}(\lambda) > 0$ for some range of k. We first consider

$$(J_{11} - k^2D_1)(J_{22} - k^2D_2) - J_{21}J_{12} = 0,$$

which will yield the critical values of k^2 for which $\mathrm{Re}(\lambda) = 0$.

The solutions to this equation are

$$k_{\pm}^2 = \frac{(D_2 J_{11} + D_1 J_{22}) \pm \sqrt{(D_2 J_{11} + D_1 J_{22})^2 - 4 D_1 D_2 \det(J)}}{2 D_1 D_2}. \tag{24}$$

Thus, in order to have $k_{\pm}^2$ positive and real and $\text{Re}(\lambda) > 0$ for $k_{-}^2 < k^2 < k_{+}^2$, we must have

$$D_2 J_{11} + D_1 J_{22} > 0, \tag{25}$$

$$(D_2 J_{11} + D_1 J_{22})^2 - 4 D_1 D_2 \det(J) > 0. \tag{26}$$

Let's first consider $D_2 J_{11} + D_1 J_{22} > 0$, keeping in mind that $\text{trace}(J) < 0$ for the stability of the ODE system. In order then for (25) to hold, we can only have either $J_{11} < 0$, or $J_{22} < 0$. Without loss of generality, we'll assume $J_{11} > 0$ and $J_{22} < 0$, and therefore $J_{22} < -J_{11}$. Combining this inequality with (25), we have that $D_2/D_1 > 1$. We see then that in order for pattern formation to occur, the diffusion coefficients cannot be equal, and more specfically, the diffusion coefficient of v must be greater than that of u.

Using (26), we can establish a more precise condition for D_2/D_1. We see that $k_{+}^2 = k_{-}^2$ when

$$(D_2 J_{11} + D_1 J_{22})^2 - 4 D_1 D_2 \det(J) = 0, \tag{27}$$

corresponding to the critical case where the spatially homogeneous solution becomes unstable. If we think about this occuring as we vary the parameter D_2/D_1, we see that a bifurcation has occurred, now in the context of the spatially extended system. Rearranging (27), we obtain the equation,

$$\frac{D_2}{D_1} - 2\sqrt{\det(J)}\sqrt{\frac{D_2}{D_1}} + \frac{J_{22}}{J_{11}} = 0,$$

which reveals the critical value of k^2,

$$k_c^2 = \sqrt{\frac{\det(J)}{D_1 D_2}}.$$

1.1.2 Summary

In establishing the conditions for pattern formation, it's easy to get caught up in the details and lose sight of the bigger picture. Let's recap the steps of the calculation:

- First, we examined the ODE system and noted the conditions that the Jacobian, J, must satisfy for the fixed point to be stable.
- Next, we took this solution as the homogeneous steady-state solution in the PDE system.

- We then linearised the PDEs about this solution to examine if perturbations will grow or decay with time.
- We established that growth or decay of the perturbations is governed by the eigenvalue, λ.
- We established conditions that J and D_2/D_1 must satisfy in order for $\mathrm{Re}(\lambda) > 0$ for some range of k.

This sequence of steps produced several conditions. In considering specific cases, it's useful to have a list of these conditions and a reminder of how they arose:

- Stability of the fixed point: $\operatorname{trace}(J) = J_{11} + J_{22} < 0$.
- Stability of the fixed point: $\det(J) = J_{11}J_{22} - J_{12}J_{21} > 0$.
- To ensure $\mathrm{Re}(\lambda) > 0$: $D_1J_{22} + D_2J_{11} > 0$.
- To ensure $\mathrm{Re}(\lambda) > 0$: $(D_2J_{11} + D_1J_{22})^2 - 4D_1D_2\det(J) > 0$

2 Pattern formation in one and two-dimensions

Let's put this theory into practise and consider the dimensionless reaction-diffusion system

$$\begin{aligned}\frac{\partial u}{\partial t} &= a - u + u^2v + \nabla^2 u\\ \frac{\partial v}{\partial t} &= b - u^2v + d\nabla^2 v,\end{aligned}$$

where the parameters a, b, and d are all positive.

Before diving into pattern formation, we must first find the fixed point of the ODE system as this will give the homogeneous steady-state about which we linearise. The system we consider here is

$$\begin{aligned}\frac{du}{dt} &= a - u + u^2v,\\ \frac{dv}{dt} &= b - u^2v,\end{aligned}$$

which as fixed point $u^* = a + b$ and $v^* = b/(a+b)^2$. We'll now require that this fixed point be stable. The Jacobian is given by

$$J(u,v) = \begin{bmatrix} 2uv-1 & u^2 \\ -2uv & -u^2 \end{bmatrix},$$

which, when evaluated at the steady state gives

$$J(u^*,v^*) = \begin{bmatrix} \frac{b-a}{a+b} & (a+b)^2 \\ -\frac{2b}{a+b} & -(a+b)^2 \end{bmatrix}. \tag{28}$$

The conditions for fixed point stability require that $J_{11} + J_{22} < 0$, giving

$$b - a < (a + b)^3.$$

Additionally we'll need $J_{11}J_{22} - J_{21}J_{12} > 0$, but this is satisfied automatically since $J_{11}J_{22} - J_{21}J_{12} = -(b - a)(a + b) + 2b(a + b) = (a + b)^2 > 0$.

Now turning our attention to the conditions for $\mathrm{Re}(\lambda) > 0$, we'll have that $D_1J_{22} + D_2J_{11} = -(a + b)^2 + d(b - a)/(a + b) > 0$, and therefore,

$$d > (a + b)^3/(b - a)$$

Additionally, since $J_{22} = -(a + b)^2 < 0$, we must have $J_{11} = (b - a)/(a + b) > 0$, meaning

$$b > a.$$

Finally, $(D_2J_{11} + D_1J_{22})^2 - 4D_1D_2\det(J) > 0$ gives

$$(d(b - a) - (a + b)^3)^2 > 4d(a + b)^4.$$

2.1 In one-dimension

Let's first consider the case of one spatial dimension, where

$$\begin{aligned}\frac{\partial u}{\partial t} &= a - u + u^2v + \frac{\partial^2 u}{\partial x^2}\\ \frac{\partial v}{\partial t} &= b - u^2v + d\frac{\partial^2 v}{\partial x^2}\end{aligned}$$

for $x \in \Omega = (0, L)$. The no-flux boundary conditions are

$$\begin{aligned}\frac{\partial u}{\partial x}(0) = 0,\ \frac{\partial u}{\partial x}(L) = 0,\\ \frac{\partial v}{\partial x}(0) = 0,\ \frac{\partial v}{\partial x}(L) = 0.\end{aligned}$$

As in the general case, linearising about the homogeneous steady-state solution yields the system

$$\frac{\partial \mathbf{W}}{\partial t} = J\mathbf{W} + \mathcal{D}\frac{\partial^2 \mathbf{W}}{\partial x^2}$$

subject to

$$\frac{\partial \mathbf{W}}{\partial x}(0) = 0, \ \frac{\partial \mathbf{W}}{\partial x}(L) = 0,$$

where, again, $\mathbf{W} = (u - u^*, v - v^*)$, J is given by (28), and $\mathcal{D}$ is the diagonal matrix with non-zero entries $\mathcal{D}_{11} = 1$, and $\mathcal{D}_{22} = d$.

As in the general case, we consider the trial solution $\mathbf{W} = \zeta F(t)\phi(x)$, which yields the two eigenvalue problems

$$\frac{dF}{dt} = \lambda F$$

and

$$\begin{aligned}&\frac{\partial^2 \phi}{\partial x^2} + k^2\phi = 0\\ &\frac{\partial \phi}{\partial x}(0) = 0, \ \frac{\partial \phi}{\partial x}(L) = 0,\end{aligned}$$

where, again, just like any eigenvalue problem, one must solve for the eigenvalues, here λ and k^2, as well as their corresponding eigenfunctions, F and ϕ, respectively.

Now that we have specified the dimension and the domain, we can solve the eigenvalue problem for ϕ to find that,

$$\phi(x) = \cos\left(\frac{n\pi x}{L}\right)$$

with eigenvalue $k = n\pi/L$, where n is a positive integer. We see, therefore that perturbations are linear combinations of cosine waves, and the eigenvalue, k, associated with each wave is proportional to the inverse of its wavelength. Since, as we established in the general case, the temporal eigenvalue λ depends on k, this tells us that perturbations with different wavelengths can grow at different rates. Others may not grow at all and instead decay away. We suspect then that the cosine wave with the highest growth rate will set the pattern.

Based on this, the most general form of the perturbation is

$$\mathbf{W}(x,t) = \sum_n a_n \boldsymbol{\zeta}_n \mathrm{e}^{\lambda(n\pi/L)t} \cos\left(\frac{n\pi x}{L}\right)$$

where the coefficients a_n would be computed to match a specific set of initial conditions. In general, we won't concern ourselves with this detail as we expect that in a biological system where stochasticity is paramount, all $a_n \neq 0$. Therefore, the final pattern would be determined primarily by the term (or mode) with the highest growth rate.

To obtain the range of k corresponding to growing (or unstable) modes, we can apply the general results established in the previous section, and compute $k_\pm^2$ as given by (24). In this case, we see that

$$\begin{aligned} D_2 J_{11} + D_1 J_{22} &= d\frac{b-a}{a+b} - (a+b)^2 \\ D_1 D_2 \det(J) &= d(a+b)^2 \end{aligned}$$

which give

$$k_-^2 = \frac{1}{2d}\left(d\frac{b-a}{a+b} - (a+b)^2 - \sqrt{\left(d\frac{b-a}{a+b} - (a+b)^2\right)^2 - 4d(a+b)^2}\right). \tag{29}$$

$$k_+^2 = \frac{1}{2d}\left(d\frac{b-a}{a+b} - (a+b)^2 + \sqrt{\left(d\frac{b-a}{a+b} - (a+b)^2\right)^2 - 4d(a+b)^2}\right). \tag{30}$$

Therefore, perturbations with spatial modes, ϕ, with k that fall within the range $k_-^2 < k^2 < k_+^2$, will grow with time and establish a pattern.

The general picture tends to get lost in the technical details of the analysis and, like many things in this course, is perhaps best illustrated graphically, or visualised through a numerical solution. The Python cell below does several things. Based on the parameter values, it computes $k_\pm^2$ to determine the range of k for which pattern formation is possible. Additionally, the code solves the full PDE system with randomly generated initial conditions that contain as many modes as is numerically viable. The resulting steady solution (i.e. pattern) achieved after a long period of evolution is plotted.

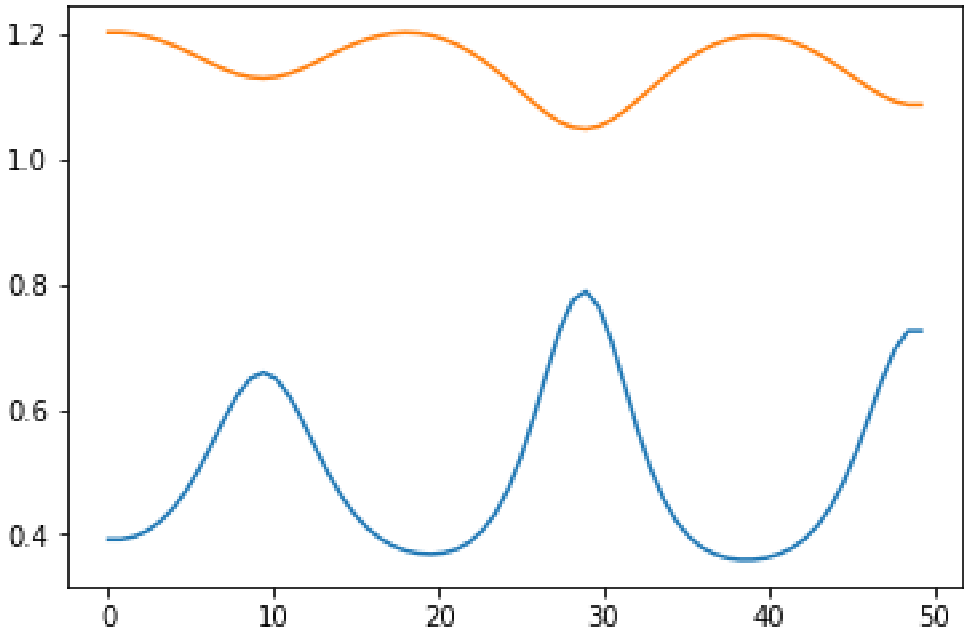
[19]:

```
## CODE FINDING A NUMERICAL SOLUTION TO THE PDE SYSTEM

import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg


a = 0.2;
b = 0.3;
d = (a + b)**2/(b - a) + 30

trace = d*(b - a)/(b + a);
det = d*(a+b)**2
```

```
ksqmin = (trace - (trace**2 - 4*det)**(0.5))/(2*d);
ksqmax = (trace + (trace**2 - 4*det)**(0.5))/(2*d);


N = 64
L = 50
dx = L/N
x = np.linspace(0, L, N, endpoint=False)

print('nmin =', L*ksqmin**0.5/np.pi)
print('nmax =', L*ksqmax**0.5/np.pi)

ustar = a + b;
vstar = b/ustar**2;

u0 = ustar + 0.01*(np.random.rand(N) - 0.5);
v0 = vstar + 0.01*(np.random.rand(N) - 0.5);

u0[0] = u0[1];
u0[N-1] = u0[N-2];

v0[0] = v0[1];
v0[N-1] = v0[N-2];

D2 = np.eye(N, k = -1) - 2*np.eye(N) + np.eye(N,k = 1)
D2 = D2/dx**2

usol0 = np.concatenate((u0, v0))

def du_dt(usol, t):
    u = usol[0:N]
    v = usol[N:2*N]
    u1 = a - u + u**2*v + D2.dot(u)
    v1 = b - u**2*v + d*D2.dot(v)

    u1[0] = u1[1];
    u1[N-1] = u1[N-2];

    v1[0] = v1[1];
    v1[N-1] = v1[N-2];

    return np.concatenate((u1, v1))

Nt = 100000
ts = np.linspace(0, 10000, Nt)
us = odeint(du_dt, usol0, ts)
```

```
plt.figure(0)
plt.plot(x,us[9999, 0:N]);
plt.plot(x,us[9999, N:2*N]);

#timeind = np.linspace(0, Nt-1, int(Nt/1000), dtype=int)
#for i in timeind:
#        plt.figure(0)
#        plt.plot(x,us[i, 0:N]);
#        plt.figure(1)
#        plt.plot(x,us[i, N:2*N]);
```

```
nmin = 3.627949193702848
nmax = 6.123608125041946
```



2.2 In two dimension

Let's now consider the two dimensional case where we have,

$$\begin{aligned}\frac{\partial u}{\partial t} &= a - u + u^2 v + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \\ \frac{\partial v}{\partial t} &= b - u^2 v + d\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right).\end{aligned}$$

We'll take the domain to be the rectangle, $\Omega = (0, L) \times (0, H)$ such that $0 \leq x \leq L$ and $0 \leq y \leq H$. The no-flux boundary conditions are

$$\begin{aligned} \frac{\partial u}{\partial x}(0, y) = 0, \ \frac{\partial u}{\partial x}(L, y) = 0, \\ \frac{\partial v}{\partial x}(0, y) = 0, \ \frac{\partial v}{\partial x}(L, y) = 0, \end{aligned}$$

as well as,

$$\begin{aligned} \frac{\partial u}{\partial y}(x, 0) = 0, \ \frac{\partial u}{\partial y}(x, H) = 0, \\ \frac{\partial v}{\partial y}(x, 0) = 0, \ \frac{\partial v}{\partial y}(x, H) = 0. \end{aligned}$$

One very nice thing to recognise is that even though we've gone to two-dimensions, much of the analysis that we've already done for the one-dimensional case carries over. The fixed points and J are the same, and accordingly, the conditions for pattern formation and the range of k^2 for which they occur will be exactly the same. What will be different will be the realised values of k now that the equation for ϕ is,

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + k^2 \phi = 0$$
$$\frac{\partial \phi}{\partial x}(0, y) = 0, \ \frac{\partial \phi}{\partial x}(L, y) = 0, \frac{\partial \phi}{\partial y}(x, 0) = 0, \ \frac{\partial \phi}{\partial y}(x, H) = 0.$$

Without going into the details (an exercise in separation of variables), the solution is

$$\phi_{n,m}(x, y) = \cos\left(\frac{n\pi x}{L}\right) \cos\left(\frac{m\pi y}{H}\right),$$

where n and m are positive integers. This solution make sense intuitively – it is the product of the 1D solution applied to the x- and y- directions separately. We see that the eigenvalue is now

$$k_{n,m}^2 = \left(\frac{n\pi}{L}\right)^2 + \left(\frac{m\pi}{H}\right)^2$$

which has separate and not necessarily equal contributions from the x- and y- directions. The general solution then is

$$\mathbf{W}(x, y, t) = \sum_n \sum_m a_{n,m} \boldsymbol{\zeta}_{n,m} \mathrm{e}^{\lambda(k_{n,m}^2)t} \cos\left(\frac{n\pi x}{L}\right) \cos\left(\frac{m\pi y}{L}\right).$$

Let's notice something. Let's take $m = 0$, which is a perfectly valid solution. This reduces the general two-dimensional solution, to the one-dimensional one! Thus, the one-dimensional solution is also a solution to the 2D equation. Similarly, we can view the 1D equation as a constrained version of the 2D one.

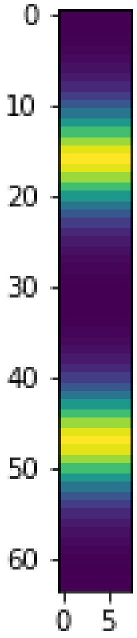
As we were saying before, many of the conditions for instability are indifferent to the underlying spatial dimension in which they are occuring. This includes the range of k for which $\mathrm{Re}(\lambda) > 0$. Thus, we must have, $k_-^2 < (n\pi/L)^2 + (m\pi/H)^2 < k_+^2$, where $k_\pm^2$ are given above in (30). While the condition is the same, we see that in 2D we have more dials to play with to satisfy it, namely the dimensions of the domain, L and H. Thus, by adjusting the geometry, we can control the patterns that emerge. In this case, we can go from stripes that occur for $H \ll L$ to a spotted pattern when H and L are comparable in size. Perhaps this is why some tabby cats have spots on their body and stripes on their tails! Both of these patterns can be explored using the Python cell below.

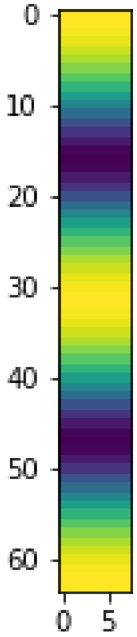
[23]:

```
# This Python cell uses a finite-difference method to integrate in time the␣
 ↪reaction-diffusion equation
# in 2D.  Please note that I have used the most basic, rudimentary, and as a␣
 ↪result, inefficient methods to
# solve this problem.  Those of you that go on to study Computational PDEs will␣
 ↪see that much can be improved
# by using a higher-order scheme to impose the boundary conditions, but more␣
 ↪importantly implicit methods and
# multigrid for time integration.


import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy.integrate import solve_ivp
from scipy import linalg


a = 0.2;
b = 0.5;
d = (a + b)**2/(b - a) + 30

trace = d*(b - a)/(b + a);
det = d*(a+b)**2

ksqmin = (trace - (trace**2 - 4*det)**(0.5))/(2*d);
ksqmax = (trace + (trace**2 - 4*det)**(0.5))/(2*d);


Nx = 64
Ny = 8;
L = 40
dx = L/Nx
```

```
H = L*Ny/Nx;

print('nmin =', L*ksqmin**0.5/np.pi)
print('nmax =', L*ksqmax**0.5/np.pi)

ustar = a + b;
vstar = b/ustar**2;

u0 = ustar + 0.01*(np.random.rand(Nx,Ny) - 0.5);
v0 = vstar + 0.01*(np.random.rand(Nx,Ny) - 0.5);

u0[0,:] = u0[1,:];
u0[Nx-1,:] = u0[Nx-2,:];
u0[:,0] = u0[:,1];
u0[:,Ny-1] = u0[:,Ny-2];

v0[0,:] = v0[1,:];
v0[Nx-1,:] = v0[Nx-2,:];
v0[:,0] = v0[:,1];
v0[:,Ny-1] = v0[:,Ny-2];

dt = 0.003
Nt = 200000
T = Nt*dt
ts = np.linspace(0, T, Nt)

u = u0[:];
v = v0[:];

fu = np.ones((Nx,Ny))
fv = np.ones((Nx,Ny))

for i in ts:

    fu[1:Nx-1,1:Ny-1] = a - u[1:Nx-1,1:Ny-1] + v[1:Nx-1,1:Ny-1]*u[1:Nx-1,1:
 ↪Ny-1]**2 + (u[0:Nx-2,1:Ny-1] + u[1:Nx-1,0:Ny-2] - 4*u[1:Nx-1,1:Ny-1] + u[2:
 ↪Nx,1:Ny-1] + u[1:Nx-1,2:Ny])/dx**2
    fv[1:Nx-1,1:Ny-1] = b - v[1:Nx-1,1:Ny-1]*u[1:Nx-1,1:Ny-1]**2 + d*(v[0:
 ↪Nx-2,1:Ny-1] + v[1:Nx-1,0:Ny-2] - 4*v[1:Nx-1,1:Ny-1] + v[2:Nx,1:Ny-1] + v[1:
 ↪Nx-1,2:Ny])/dx**2

    fu[0,:] = fu[1,:];
    fu[Nx-1,:] = fu[Nx-2,:];
    fu[:,0] = fu[:,1];
    fu[:,Ny-1] = fu[:,Ny-2];

    fv[0,:] = fv[1,:];
    fv[Nx-1,:] = fv[Nx-2,:];
    fv[:,0] = fv[:,1];
    fv[:,Ny-1] = fv[:,Ny-2];

    u += dt*fu;
    v += dt*fv;

#print('here')
plt.figure(0)
plt.imshow(u);
plt.figure(1)
plt.imshow(v);
```

```
nmin = 2.541648814661189
nmax = 7.938350229862815
```




0
10
20
30
40
50
60
0
5

[ ]:

Mathematical Biology - Week 9

November 29, 2022

1 Stochastic processes

In the final part of the course, we'll discuss stochastic processes that capture the inherent random nature of biological phenomena. In these models, the quantity of interest will often be the probability that our system is of a certain size. For example, a question that one might ask of such a model is: what is the probability that the human population in 2138 will be 332342356 given its current size? Based on this, it is interesting to consider how these models are related to the deterministic ODE models that we discussed earlier, and whether similar predictions emerge. For example, again regarding population dynamics, we might ask does: the ODE model describe the dynamics of the mean population size of the related stochastic process?

1.1 Continuous-Time Markov Chains (CTMC)

The models that we will discuss fall under the category of *continuous-time Markov chains*, which we will introduce here. Suppose the quantity $X(t)$ is a random variable that depends on time, t, and takes values in the natural numbers, $0, 1, 2, \ldots$. For example, $X(t)$ could represent the size of a population at time t, or perhaps the number of molecules involved in a chemical reaction. We'll be interested in answering the question: what is $p_i(t) = \text{Prob}\{X(t) = i\}$, the probability that $X = i$ at time t? Once we know $p_i(t)$ for each i, we can compute the mean of the process through

$$m(t) = \sum_{i=0}^{\infty} i p_i(t),$$

as well as the variance,

$$\sigma^2(t) = \sum_{i=0}^{\infty} i^2 p_i(t) - m^2(t).$$

1.1.1 Transition probabilities

The answer to the question posed above will often depend on the state of the system at previous times. Thinking again about a growing population – the population size at time t is more likely to be larger if it had a greater value at time s for some $s < t$. Thus, what we would like to do is relate $X(t)$ to $X(s)$. This is done via the *transition probability*

$$p_{ji}(t-s) = \text{Prob}\left\{X(t) = j | X(s) = i\right\}$$

for $s < t$. Here, we have assumed that the process is stationary, i.e. $p_{ji}(t,s) = p_{ji}(t-s)$. The transition probabilities can be viewed as the entries of a matrix, $P(t) = [p_{ji}(t)]$. The transition probabilities also satisfy what is known as the Chapman-Kolmogorov equations

$$p_{ji}(t+s) = \sum_{k=0}^{\infty} p_{jk}(t) p_{ki}(s),$$

which, in matrix form is, $P(t+s) = P(t)P(s)$. And, in most cases, (in fact, for all the cases that we will consider) we will also have,

$$\sum_{j=0}^{\infty} p_{ji}(t) = 1, t \geq 0$$

Infinitesimal transition probabilities for the Poisson process One of the most fundamental stochastic processes is the Poisson process. One way to define the Poisson process is by specifying the infinitesimal transition probabilities, $p_{ji}(\Delta t)$ for $\Delta t \ll 1$. A Poisson process is defined as

1. At $t = 0$, $X(0) = 0$.
2. For sufficiently small Δt, the transition probabilities are given by

$$\begin{aligned} p_{i+1,i}(\Delta t) &= \text{Prob}\left\{X(t+\Delta t) = i+1 | X(t) = i\right\} = \lambda \Delta t + o(\Delta t) \\ p_{ii}(\Delta t) &= \text{Prob}\left\{X(t+\Delta t) = i | X(t) = i\right\} = 1 - \lambda \Delta t + o(\Delta t) \\ p_{ji}(\Delta t) &= \text{Prob}\left\{X(t+\Delta t) = j | X(t) = i\right\} = o(\Delta t), j \geq i+2 \\ p_{ji}(\Delta t) &= 0, j < i \end{aligned}$$

where $o(\Delta t)$ (‘’little oh Δt”) indicates quantites that are smaller than Δt in the limit $\Delta t \to 0$, i.e. $f(\Delta t) = o(\Delta t)$ if

$$\lim_{\Delta t \to 0} \frac{f(\Delta t)}{\Delta t} = 0$$

1.1.2 Generator matrix

Let's assume that the entries $p_{ji}(t)$ are continuously differentiable for $t \geq 0$ and that for $t = 0$, they satisfy $p_{ji}(0) = 0$ for $j \neq i$, and $p_{ii}(0) = 1$. The entries of the generator matrix, q_{ji}, are defined as

$$q_{ji} = \lim_{\Delta t \to 0^+} \frac{p_{ji}(\Delta t) - p_{ji}(0)}{\Delta t} = \lim_{\Delta t \to 0^+} \frac{p_{ji}(\Delta t)}{\Delta t}$$

for $i \neq j$, and

$$q_{ii} = \lim_{\Delta t \to 0^+} \frac{p_{ji}(\Delta t) - p_{ji}(0)}{\Delta t} = \lim_{\Delta t \to 0^+} \frac{p_{ji}(\Delta t) - 1}{\Delta t}.$$

Since $\sum_{j=0}^{\infty} p_{ji}(\Delta t) = 1$, we have $1 - p_{ii}(\Delta t) = \sum_{j \neq i} p_{ji}(\Delta t)$ and, therefore,

$$q_{ii} = -\sum_{j \neq i} q_{ji}.$$

Accordingly, for sufficiently small we can express the transition probabilities as

$$p_{ji}(\Delta t) = \delta_{ji} + q_{ji}\Delta t + o(\Delta t), \tag{1}$$

and in matrix form as $P(\Delta t) = I + Q\Delta t + o(\Delta t)$, where $Q = [q_{ji}]$.

1.1.3 Generator matrix for the Poisson process

Based on the transition probabilities for the Poisson process, we see that the non-zero entries of the generator matrix are

$$\begin{aligned} q_{ii} &= -\lambda \\ q_{i+1,i} &= \lambda \end{aligned}$$

1.2 Kolmogorov differential equations

We can put together what we have done so far to derive a differential equation for $p_i(t)$. Let's first consider the Chapman-Kolmogorov equation,

$$p_{ji}(t + \Delta t) = \sum_{k=0}^{\infty} p_{jk}(\Delta t) p_{ki}(t)$$

and, using (1) from above write,

$$p_{ji}(t+\Delta t) = \sum_{k=0}^{\infty} \left[\delta_{jk} + q_{jk}\Delta t + o(\Delta t)\right] p_{ki}(t)$$

Since $\sum_{k=0}^{\infty} \delta_{jk} p_{ki}(t) = p_{ji}(t)$, and $\sum_{k=0}^{\infty} p_{ki}(t) = 1$, we have

$$p_{ji}(t+\Delta t) = p_{ji}(t) + \Delta t \sum_{k=0}^{\infty} q_{jk} p_{ki}(t) + o(\Delta t).$$

Substracting $p_{ji}(t)$ from both sides, dividing by through by Δt, and taking $\Delta t \to 0$ gives

$$\frac{dp_{ji}}{dt} = \sum_{k=0}^{\infty} q_{jk} p_{ki}(t),$$

which in matrix form is

$$\frac{dP}{dt} = QP.$$

This is what is known as the *forward Kolmogorov differential equation*, or the *master equation*.

Now, suppose that $X(0) = k$. Then the state probability $p_i(t)$ will be the same as the transition probability $p_{ik}(t) = \text{Prob}\left\{X(t) = i | X(0) = k\right\}$. Thus, we can set p as the kth column of P and the forward Kolmogorov equation gives

$$\frac{dp}{dt} = Qp,$$

or in component form

$$\frac{dp_i}{dt} = \sum_{k=0}^{\infty} q_{ik} p_k(t). \tag{2}$$

Forward Kolmogorov equation for the Poisson process Based on the definition of the Poisson process, we'll have $p_i(t) = p_{i0}(t)$, and using the generator matrix for the Poisson process, the forward Kolmogorov differential equation gives

$$\begin{aligned} \frac{dp_0}{dt} &= -\lambda p_0 \\ \frac{dp_i}{dt} &= \lambda p_{i-1} - \lambda p_i, i \geq 1 \end{aligned}$$

These resulting equations for $p_i(t)$ can be solved successively. The initial condition that $X(0) = 0$ tells us that $p_0(0) = 1$, while $p_i(0) = 0$ for $i > 0$. The differential equation for p_0 gives

$$p_0(t) = e^{-\lambda t}.$$

Thus, for $p_1(t)$, we have

$$\frac{dp_1}{dt} = \lambda e^{-\lambda t} - \lambda p_1,$$

whose solution (after applying the initial condition) is

$$p_1(t) = \lambda t e^{-\lambda t}.$$

Continuing on, for $p_2(t)$, we have

$$\frac{dp_2}{dt} = \lambda^2 t e^{-\lambda t} - \lambda p_2,$$

which, with $p_2(0) = 0$, has solution

$$p_2(t) = \frac{(\lambda t)^2}{2!} e^{-\lambda t}.$$

Carrying the process forward, one finds in general that

$$p_i(t) = \frac{(\lambda t)^i}{i!} e^{-\lambda t}.$$

With $p_i(t)$ in hand, we can compute the mean,

$$\begin{aligned} m(t) &= \sum_{i=0}^{\infty} i \frac{(\lambda t)^i}{i!} e^{-\lambda t}, \\ &= \lambda t e^{-\lambda t} \sum_{i=1}^{\infty} \frac{(\lambda t)^{i-1}}{(i-1)!}, \\ &= \lambda t, \end{aligned}$$

as well as the variance,

$$\begin{aligned}\sigma^2(t) &= \sum_{i=0}^{\infty} i^2 \frac{(\lambda t)^i}{i!} e^{-\lambda t} - (\lambda t)^2 \\ &= e^{-\lambda t}\left[\lambda t \sum_{i=1}^{\infty}(i-1)\frac{(\lambda t)^{i-1}}{(i-1)!} + \lambda t \sum_{i=1}^{\infty}\frac{(\lambda t)^{i-1}}{(i-1)!}\right] - (\lambda t)^2 \\ &= \lambda t\end{aligned}$$

1.2.1 Generating functions

Rather than solving for the $p_i(t)$ directly, an alternative approach is to use a *generating function* of the form

$$\mathcal{P}(z,t) = \sum_{i=0}^{\infty} p_i(t) z^i. \tag{3}$$

This may seem like an odd thing to do, but notice that

$$\begin{aligned} p_i(t) &= \frac{1}{i!}\left.\frac{\partial^i \mathcal{P}}{\partial z^i}\right|_{z=0} && (4) \\ m(t) &= \left.\frac{\partial \mathcal{P}}{\partial z}\right|_{z=1}, && (5) \\ \sigma^2(t) &= \left.\frac{\partial^2 \mathcal{P}}{\partial z^2}\right|_{z=1} + \left.\frac{\partial \mathcal{P}}{\partial z}\right|_{z=1} - \left(\left.\frac{\partial \mathcal{P}}{\partial z}\right|_{z=1}\right)^2. && (6)\end{aligned}$$

Thus, once we know the generating function, we can extract information about the moments of the process, as well as values of $p_i(t)$.

We can derive a PDE for $\mathcal{P}(z,t)$ using the forward Kolmogorov equation. If we differentiate (3) with respect to time, we have

$$\frac{\partial \mathcal{P}}{\partial t} = \sum_{i=0}^{\infty} \frac{dp_i}{dt} z^i.$$

Using (2), we arrive at

$$\frac{\partial \mathcal{P}}{\partial t} = \sum_{k=0}^{\infty}\sum_{i=0}^{\infty} q_{ik} p_k z^i. \tag{7}$$

The trick will now be to relate the sum on the right hand side back to $\mathcal{P}(z,t)$ and its derivatives with respect to z, thereby generating the PDE. This will vary from process to process.

Generating function for the Poisson process To see how this works, let's consider again the Poisson process. As when we applied the Chapman-Kolmogorov equations directly, we have

$$\begin{aligned}\sum_{k=0}^{\infty} q_{0k} p_k &= -\lambda p_0 \\ \sum_{k=0}^{\infty} q_{ik} p_k &= -\lambda p_i + \lambda p_{i-1}, \text{ for i} > 0.\end{aligned}$$

Thus, from (7), we have

$$\frac{\partial \mathcal{P}}{\partial t} = -\lambda \sum_{i=0}^{\infty} p_i(t) z^i + \lambda \sum_{i=1} p_{i-1}(t) z^i.$$

Writing

$$\sum_{i=1} p_{i-1}(t) z^i = z \sum_{i=0} p_i(t) z^i,$$

gives

$$\frac{\partial \mathcal{P}}{\partial t} = -\lambda \mathcal{P} + \lambda z \mathcal{P} = \lambda(z-1)\mathcal{P}$$

whose general solution is

$$\mathcal{P}(z,t) = \mathcal{P}(z,0) e^{\lambda(z-1)t}.$$

Since $p_0(0) = 1$ and $p_i(0) = 0$ for $i > 0$, it follows that $\mathcal{P}(z,0) = 1$. Thus,

$$\mathcal{P}(z,t) = e^{\lambda(z-1)t}.$$

The values of $m(t)$ and $\sigma^2(t)$ obtained previously can be confirmed here by applying (6).

1.3 Simple birth process

With the tools above, we can begin to analyse the stochastic process analogues of the ODE models that we studied earlier in the course. We begin with the simple birth process for which we have $X(0) = N$, and transition probabilities given by

$$p_{i+j,i}(\Delta t) = \text{Prob}\left\{X(t+\Delta t) = i+j | X(t) = i\right\} \tag{8}$$

$$= \begin{cases} \lambda i \Delta t + o(\Delta t), & j = 1 \\ 1 - \lambda i \Delta t + o(\Delta t), & j = 0 \\ o(\Delta t), & j \geq 2 \\ 0, & j < 0. \end{cases} \tag{9}$$

Thus, if we have a population of size i at time t, then the probability it will grow to a size $i+1$ by $t + \Delta t$ is given by $\lambda i \Delta t + o(\Delta t)$. We see then that λ is the birth rate per capita, just as we had before with our deterministic ODE model. Naturally then, we would like to explore how the two models are related, and see what additional information, if any, might be captured by taking into account the stochastic nature of birth events.

1.3.1 Mean and variance

For the simple birth process, we can compute equations for the mean and variance in a relatively straightforward manner using the forward Kolomogorov equation. Examining (9), we see that the non-zero entries of the generator matrix are given by

$$\begin{aligned} q_{ii} &= -\lambda i, \quad i \geq N, \\ q_{i+1,i} &= \lambda i, \quad i \geq N \end{aligned}$$

and, therefore, the forward Kolomogorov equation is

$$\begin{aligned} \frac{dp_{Nj}}{dt} &= -\lambda N p_{Nj}, \\ \frac{dp_{ij}}{dt} &= \lambda(i-1)p_{i-1,j} - \lambda i p_{ij}, \; i \geq N+1. \end{aligned}$$

Since we have $X(0) = N$, the state probability is given by $p_i(t) = p_{iN}(t)$ and, therefore,

$$\begin{aligned} \frac{dp_N}{dt} &= -\lambda N p_N, \\ \frac{dp_i}{dt} &= \lambda(i-1)p_{i-1} - \lambda i p_i, \; i \geq N+1. \end{aligned}$$

To find the mean, we multiply these equations by i, and summing over i, we have

$$\sum_{i=N}^{\infty} i\frac{dp_i}{dt} = \lambda \sum_{i=N+1}^{\infty} i(i-1)p_{i-1} - \sum_{i=N}^{\infty} \lambda i^2 p_i. \tag{10}$$

Since,

$$\sum_{i=N+1}^{\infty} i(i-1)p_{i-1} = \sum_{i=N}^{\infty} i(i+1)p_i,$$

we have

$$\begin{aligned}\lambda\left(\sum_{i=N+1}^{\infty} i(i-1)p_{i-1} - \sum_{i=N}^{\infty} i^2 p_i\right) &= \lambda\left(\sum_{i=N}^{\infty}(i+1)ip_i - \sum_{i=N}^{\infty} i^2 p_i\right)\\ &= \lambda\sum_{i=N}^{\infty} ip_i\\ &= \lambda m.\end{aligned}$$

As,

$$\begin{aligned}\sum_{i=N}^{\infty} i\frac{dp_i}{dt} &= \frac{d}{dt}\sum_{i=N}^{\infty} ip_i\\ &= \frac{dm}{dt},\end{aligned}$$

the equation (10) becomes

$$\frac{dm}{dt} = \lambda m,$$

which has general solution $m(t) = m(0)e^{\lambda t}$. As $m(0) = \sum_i ip_i(0) = N$,

$$m(t) = Ne^{\lambda t}.$$

We see then, that the equation for the mean of the simple birth process is identical to the determinisitic ODE model that we posed at the beginning of the course.

The additional information then that the stochasic model will provide is about the probability distribution of the population, allowing us to answer the question: how confident can we be in saying that the mean gives an accurate estimate of the population size?

This question can be answered, in part, by knowing the variance, which will give some measure of the width of the probability distribution of the population. For this simple process, we can follow a similar procedure as we had done for the mean. However, let's instead take the route of the generating function as this technique will prove more useful when we are dealing with a more complex process.

1.3.2 Generating function for the simple birth process

Putting together (7), the expressions for the state probabilities, and generator matrix for the simple birth process, we have that

$$\frac{\partial \mathcal{P}}{\partial t} = \lambda \sum_{i=N+1}^{\infty} (i-1) p_{i-1}(t) z^i - \lambda \sum_{i=N}^{\infty} i p_i(t) z^i. \tag{11}$$

We can rewrite the first sums on the righthand side as

$$\begin{aligned} \lambda \sum_{i=N+1}^{\infty} (i-1) p_{i-1}(t) z^i &= \lambda \sum_{i=N}^{\infty} i p_i(t) z^{i+1} \\ &= \lambda z^2 \sum_{i=N}^{\infty} i p_i(t) z^{i-1}. \end{aligned}$$

and

$$\lambda \sum_{i=N}^{\infty} i p_i(t) z^i = \lambda z \sum_{i=N}^{\infty} i p_i(t) z^{i-1}.$$

Since $\mathcal{P} = \sum_{i=N}^{\infty} p_i(t) z^i$, we have $\partial \mathcal{P}/\partial z = \sum_{i=N}^{\infty} i p_i(t) z^{i-1}$ and, therefore (11) gives,

$$\frac{\partial \mathcal{P}}{\partial t} = \lambda z(z-1) \frac{\partial \mathcal{P}}{\partial z}. \tag{12}$$

We'll be interested in solving this equation subject to the initial condition that $\mathcal{P}(z,0) = z^N$.

Solving this problem is an exercise in the *method of characteristics*. With the method of characterisitics, the variables t and z are assumed to be parametrised by variables τ and s such that $z(s,\tau)$ and $t(s,\tau)$. Then, along the curves where $s =$ constant, we have

$$\frac{d\mathcal{P}}{d\tau} = \frac{\partial \mathcal{P}}{\partial t}\frac{dt}{d\tau} + \frac{\partial \mathcal{P}}{\partial z}\frac{dz}{d\tau}.$$

Taking the $d\mathcal{P}/d\tau$ to be given by the PDE (12) and comparing with the equation above, we see that

$$\begin{aligned}\frac{d\mathcal{P}}{d\tau} &= 0\\ \frac{dt}{d\tau} &= 1\\ \frac{dz}{d\tau} &= \lambda z(1-z)\end{aligned}$$

which after integration yield,

$$\begin{aligned}\mathcal{P} &= c_1 && (13)\\ t &= \tau + c_2 && (14)\\ z &= \frac{c_3 e^{\lambda\tau}}{1+c_3 e^{\lambda\tau}}. && (15)\end{aligned}$$

The next step is to consider $\tau = 0$, and insist that the data provided, $\mathcal{P}(0,z) = z^N$, is along a curve parametrised by s. As that data is along the z-axis, the curve is given by $t = 0$, and $z = s$. Using these expressions and the fact that $\tau = 0$ in (15), we have that $c_1 = s^N$, $c_2 = 0$ and $c_3 = s/(1-s)$. Thus, $\tau = t$, and rearranging the expression for z in (15), we also have

$$s = \frac{ze^{-\lambda t}}{1 - z(1-e^{-\lambda t})},$$

and,

$$\mathcal{P}(z,t) = \frac{z^N e^{-N\lambda t}}{[1 - z(1-e^{-\lambda t}]^N}.$$

With this expression, we can now use (6) to compute the variance,

$$\sigma^2(t) = Ne^{2\lambda t}(1-e^{-\lambda t}).$$

We see then that as the population grows, so too does the width of the distribution. In fact, we see that it grows exponentially, however, it remains constant relative to the mean,

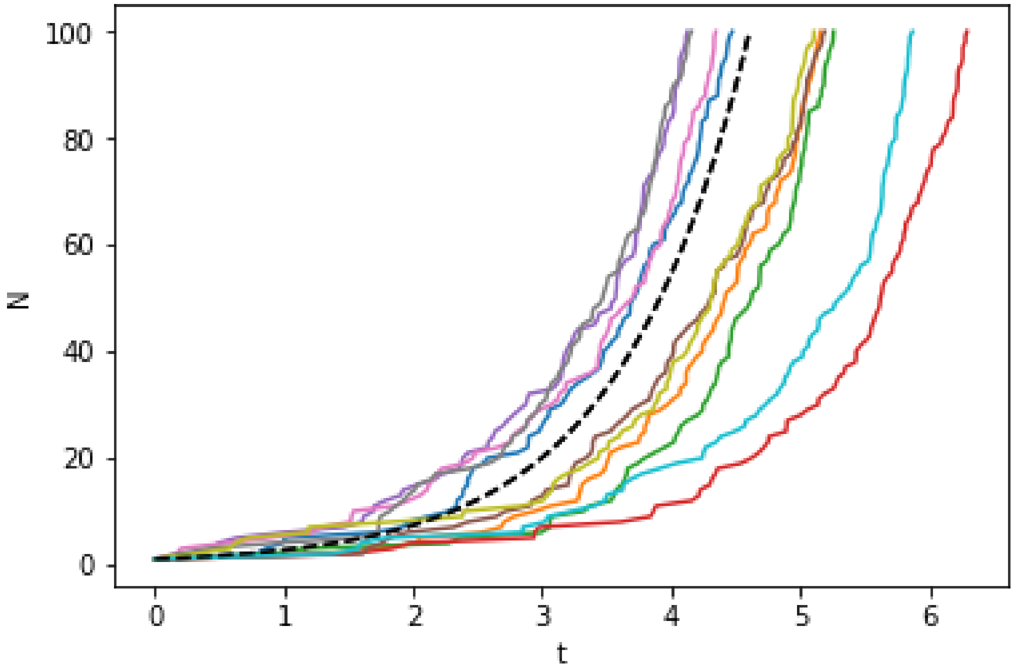
$$\lim_{t\to\infty}\frac{\sigma(t)}{m(t)} = 1/\sqrt{N}.$$

1.3.3 Sample paths via Gillespie's algorithm

A direct numerical approach to obtain sample trajectories is known as Gillespie's algorithm, which is suitable when the population sizes remain relatively small. For a simple birth process, Gillespie's algorithm involves simply generating a random variable and from it, computing the time that the next birth event occurs. The code in the cell below generates sample paths of the simple birth process.

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

lam = 1.0
x0 = 1
xe = 100
t = np.linspace(0, np.log(100)/lam, 100)
mean = x0*np.exp(lam*t)
n = np.arange(100) + 1;
nind = np.arange(99);
nsamp = np.arange(10);
tgill = np.zeros(100);

plt.plot(t,mean,'k--')

tsum = np.zeros(100);

for j in nsamp:
    tgill[0] = 0;
    for i in nind:
        tgill[i+1] = tgill[i] - np.log(np.random.rand(1,1))/(lam*n[i])
    plt.plot(tgill, n);

plt.plot(t,mean,'k--');
plt.xlabel('t')
plt.ylabel('N');
```



1.4 Simple Birth and Death process

In our initial studies population dynamics, we allowed for a constant birth rate per capita, λ, and death rate per capita, μ. If $\lambda - \mu > 0$, then we found that the population size will grow exponentially, while if $\lambda - \mu < 0$, the population will decay exponentially. Let's now include death into the simple birth process to examine these competing effects in the stochastic case.

To add death to the simple birth process, we allow for transition probabilities from higher population sizes to lower sizes with rate per capita μ. Thus, the transition probabilities are

$$p_{i+j,i}(\Delta t) = \text{Prob}\left\{X(t+\Delta t) = i+j | X(t) = i\right\} \tag{16}$$

$$= \begin{cases} \lambda i \Delta t + o(\Delta t), & j = 1 \\ 1 - (\lambda + \mu) i \Delta t + o(\Delta t), & j = 0 \\ \mu i \Delta t + o(\Delta t), & j = -1 \\ o(\Delta t), & j \neq -1, 0, 1. \end{cases} \tag{17}$$

We'll again take $X(0) = N$. From the transition probabilities, we see that non-zero entries of the generator matrix will be

$$\begin{aligned} q_{ii} &= -(\lambda + \mu) i, \\ q_{i+1,i} &= \lambda i, \\ q_{i-1,i} &= \mu i. \end{aligned}$$

Thus, the inclusion of death into the model results in a tridiagonal generator matrix. It also allows the population to shrink below the original size, something that was not possible when we only had births.

As the initial population size is N, the state probability is $p_i(t) = p_{iN}(t)$ and the forward Kolmogorov differential equation gives,

$$\frac{dp_0}{dt} = \mu p_1(t) \tag{18}$$

$$\frac{dp_i}{dt} = -(\lambda + \mu) i p_i(t) + \mu(i+1)p_{i+1}(t) + \lambda(i-1)p_{i-1}(t). \tag{19}$$

At this stage, it is important to notice that if the population reaches $X(t^*) = 0$ at some time t^*, then $X(t) = 0$ for $t \geq t^*$. This means that $X = 0$ is an *absorber*. It also means that the vector $\pi = (1, 0, 0, ...)$ is the unique *stationary probability distribution* as it satisfies

$$Q\pi = 0, \ \sum_{i=0}^{\infty} \pi_i = 1, \text{ and } \pi_i \geq 0$$

for $i = 0, 1, 2, \ldots$.

For the simple birth and death process, computing directly $p_i(t)$ is not as straightforward as it was in the case of the simple birth process. We will immediately switch our attention to finding the generating function. As usual, we multiply (19) by z^i and sum over i to find,

$$\frac{\partial \mathcal{P}}{\partial t} = -(\lambda + \mu) \sum_i i p_i(t) z^i + \mu \sum_i (i+1) p_{i+1}(t) z^i - \lambda \sum_i (i-1) p_{i-1}(t) z^i.$$

The sums in the first and third terms are identical to those we considered in the simple birth process. The sum in the second term can be written as,

$$\begin{aligned} \sum_i (i+1) p_{i+1}(t) z^i &= \sum_i i p_i(t) z^{i-1} \\ &= \frac{\partial \mathcal{P}}{\partial z}. \end{aligned}$$

Thus, the generating function for $\lambda \neq \mu$ is the solution to the partial differential equation

$$\frac{\partial \mathcal{P}}{\partial t} = (z-1)(\lambda z - \mu) \frac{\partial \mathcal{P}}{\partial z},$$

with initial condition $\mathcal{P}(z, 0) = z^N$. We can solve this equation using the method of characteristics. Along the curves of constant s, we have

$$
\begin{aligned}
\frac{d\mathcal{P}}{d\tau} &= 0 \\
\frac{dt}{d\tau} &= 1 \\
\frac{dz}{d\tau} &= (1-z)\left(\lambda z - \mu\right)
\end{aligned}
$$

and therefore,

$$
\begin{aligned}
\mathcal{P} &= c_1 && (20) \\
t &= \tau + c_2 && (21) \\
\frac{z-\mu/\lambda}{1-z} &= c_3 e^{(\lambda-\mu)\tau}. && (22)
\end{aligned}
$$

Taking the intial data and the curve along which it is given as being parametrised by s, we have that

$$
\begin{aligned}
t(s,0) &= 0, \\
z(s,0) &= s, \\
P(s,0) &= s^N
\end{aligned}
$$

and therefore, $t = \tau$ ($c_2 = 0$), and

$$c_3 = \frac{s-\mu/\lambda}{1-s}.$$

Using these expressions in the equation for z in (22), we have

$$s = \frac{e^{(\mu-\lambda)t}(\lambda z - \mu) - \mu(z-1)}{e^{(\mu-\lambda)t}(\lambda z - \mu) - \lambda(z-1)}$$

and thus,

$$\mathcal{P}(z,t) = \left(\frac{e^{(\mu-\lambda)t}(\lambda z - \mu) - \mu(z-1)}{e^{(\mu-\lambda)t}(\lambda z - \mu) - \lambda(z-1)}\right)^N, \lambda \neq \mu.$$

We can now go ahead and differentiate $\mathcal{P}$ to compute the mean and variance and find that

$$
\begin{aligned}
m(t) &= Ne^{(\lambda-\mu)t},\\
\sigma^2(t) &= N\frac{\lambda+\mu}{\lambda-\mu}e^{(\lambda-\mu)t}\left(e^{(\lambda-\mu)t}-1\right).
\end{aligned}
$$

We see then, that like the simple birth process, the mean satisfies the ODE,

$$\frac{dm}{dt} = (\lambda-\mu)t,$$

leading to exponential growth if $\lambda > \mu$, or exponential decay for $\lambda < \mu$. This again coincides with our ODE model from earlier in the course.

It may seem then that stochastic model doesn't provide much additional information, but let's look more closely. Even though we were not able to compute $p_i(t)$ directly, we can do so from the generating function using,

$$p_i(t) = \frac{1}{i!}\left.\frac{\partial^i \mathcal{P}}{\partial z^i}\right|_{z=0},$$

and therefore, we have

$$p_0(t) = \left(\frac{\mu-\mu e^{(\mu-\lambda)t}}{\lambda-\mu e^{(\mu-\lambda)t}}\right)^N.$$

If we let $t \to \infty$,

$$p_0(\infty) = \begin{cases} 1, & \text{if}\lambda \le \mu \\ \left(\frac{\mu}{\lambda}\right)^N, & \text{if}\lambda > \mu. \end{cases}$$

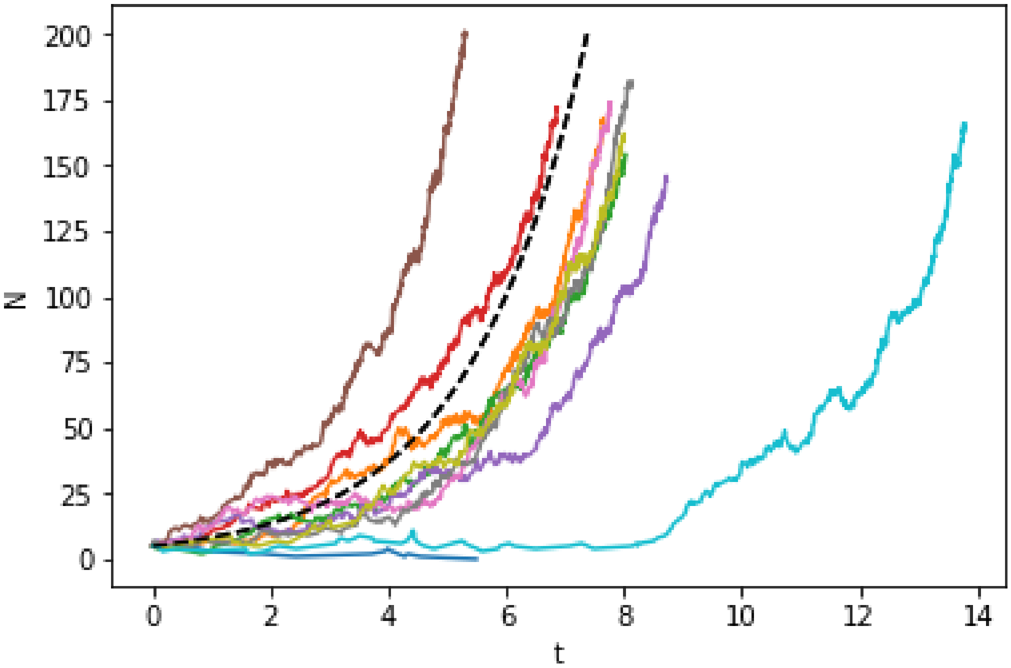
We see then that even though the mean might be growing, any particular trajectory has a finite probability that it will go to zero as $t \to \infty$. This probability depends on the ratio of death to birth rates, as well as the initial population size. Let's consider some numerical values to put this into context. If $\mu/\lambda = 0.9$ and $N = 1000$, then $p_0(\infty) = 1.7 \times 10^{-46}$, so the chances are slim! If instead, $N = 50$, then $p_0 = 5.2 \times 10^{-3}$. This shows the importance of the stochastic model when the population size is rather small.

The cell below uses Gillespie's algorithm to generate sample trajectories of the simple birth and death process and indeed one can find cases where the population becomes extinct even though $\lambda > \mu$.

[2]:

```python
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

lam = 1
mu = 0.5
x0 = 5
xe = 100
t = np.linspace(0, np.log(40)/(lam - mu), 200)
mean = x0*np.exp((lam - mu)*t)
nind = np.arange(499);
nsamp = np.arange(10);
tgill = np.zeros(500);
n = np.zeros(500);
plt.plot(t,mean,'k--')

for j in nsamp:
    tgill[0] = 0;
    n[0] = 5;
    for i in nind:
        tgill[i+1] = tgill[i] - np.log(np.random.rand(1,1))/((lam + mu)*n[i])
        bord = np.random.rand(1,1);
        if bord < lam/(lam + mu):
            n[i+1] = n[i] + 1
        else:
            n[i+1] = n[i] - 1
    plt.plot(tgill, n);

plt.plot(t,mean,'k--');

plt.xlabel('t')
plt.ylabel('N');
```

```
/Users/ekeaveny/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:23:
RuntimeWarning: divide by zero encountered in true_divide
```


200
175
150
125
100
75
50
25
0
N
0
2
4
6
8
10
12
14
t

Mathematical Biology - Week 10

December 17, 2022

0.1 Stationary Probability Distribution for a Birth and Death process

A concept that surfaced in studying simple birth and death processes is that of the stationary probability distribution – the fixed point of the forward Kolmogorov equations. For the simple birth and death process, we saw that this was given by $\pi = (1, 0, 0, \dots)^T$. What can we say about the stationary probability distribution for a *general* birth and death process?

The transition probabilties for the general birth and death process are

$$
\begin{aligned}
p_{i+j,i}(\Delta t) &= \text{Prob}\left\{X(t+\Delta t) = i+j | X(t) = i\right\} \quad (1)\\
&= \begin{cases} \lambda_i \Delta t + o(\Delta t), & j = 1 \\ 1 - (\lambda_i + \mu_i)\Delta t + o(\Delta t), & j = 0 \\ \mu_i \Delta t + o(\Delta t), & j = -1 \\ o(\Delta t), & j \neq -1, 0, 1. \end{cases} \quad (2)
\end{aligned}
$$

where $\lambda_i \geq 0$ and $\mu_i \geq 0$ for $i = 0, 1, 2, \dots$ and $\mu_0 = 0$. For our simple birth and death process, we had $\lambda_i = \lambda i$ and $\mu_i = \mu i$. From the transition probabilities above, we see that the entries of the generator matrix will be

$$
\begin{aligned}
q_{ii} &= -(\mu_i + \lambda_i) \\
q_{i+1,i} &= \lambda_i \\
q_{i-1,i} &= \mu_i
\end{aligned}
$$

and, therefore,

$$
Q = \begin{bmatrix} -\lambda_0 & \mu_1 & & & \\ \lambda_0 & -(\mu_1 + \lambda_1) & \mu_2 & & \\ & \lambda_1 & -(\mu_2 + \lambda_2) & \mu_3 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}.
$$

Recall that the stationary probability distribution is given by, $Q\pi = 0$ with $\sum_{i=0}^{\infty} \pi_i = 1$ and $\pi_i \geq 0$ for $i = 1, 2, \dots$. Thus, evaluating $Q\pi = 0$ for the general birth process, we have

$$\begin{aligned}0 =& -\lambda_0\pi_0 + \mu_1\pi_1\\ 0 =& \lambda_{i-1}\pi_{i-1} - (\lambda_i + \mu_i)\pi_i + \mu_{i+1}\pi_{i+1}.\end{aligned}$$

This can be solve recursively to find

$$\begin{aligned}\pi_1 &= \frac{\lambda_0}{\mu_1}\pi_0,\\ \pi_2 &= \frac{\lambda_0\lambda_1}{\mu_1\mu_2}\pi_0,\\ &\vdots\\ \pi_i &= \frac{\lambda_0\lambda_1\cdots\lambda_{i-1}}{\mu_1\mu_2\cdots\mu_i}\pi_0,\end{aligned}$$

Then, using the constraint that

$$\sum_{i=0}^{\infty}\pi_i = \pi_0\left(1 + \sum_{i=1}^{\infty}\frac{\lambda_0\lambda_1\cdots\lambda_{i-1}}{\mu_1\mu_2\cdots\mu_i}\right) = 1,$$

one can solve for π_0.

To get an idea of how this works, consider the birth and death process where $\lambda_i = b$ and $\mu_i = id$, where $i = 0, 1, \ldots$ and $b, d > 0$. In this case,

$$\frac{\lambda_0\lambda_1\cdots\lambda_{i-1}}{\mu_1\mu_2\cdots\mu_i} = \left(\frac{b}{d}\right)^i\frac{1}{i!}.$$

Since,

$$e^{b/d} = 1 + \sum_{i=1}^{\infty}\frac{1}{i!}\left(\frac{b}{d}\right)^i,$$

we have

$$\pi_i = \left(\frac{b}{d}\right)^i\frac{1}{i!}e^{-b/d}$$

and thus, the stationary distribution is a Poisson process with λt replaced by the parameter ratio b/d.

0.2 Logistic growth process

Recall from our discussion of exponential growth or decay, that as a model, it fails to account for the fact that the population relies on resources from its surrounding, and as a results should be maintained at a particular level. We saw that these effects are incorporated in the logistic equation,

$$\frac{dn}{dt} = rn\left(1 - \frac{n}{K}\right),$$

which emits $n^* = 0$ and $n^* = K$ (the carrying capacity) as fixed points, with the finite population size $n^* = K$ being stable.

Let's go ahead now and begin formulating the logistic growth process. Notice that the right hand side of the logistic equation indicates the difference between the birth and death rates. Thus, for the logistic growth process, we must have

$$\lambda_n - \mu_n = rn - \frac{r}{K}n^2, \tag{3}$$

which stipulates that the the birth and death rates are quadratic in the population size. Assuming the population can grow without bound (the state space is infinite), the birth and death rates are

$$\lambda_i = b_1 i + b_2 i^2 > 0 \tag{4}$$
$$\mu_i = d_1 i + d_2 i^2 > 0 \tag{5}$$

where b_1, d_1, b_2, and d_2 are positive constants. Combining this with (3), we see that

$$r = b_1 - d_1 > 0, \text{ and } K = \frac{b_1 - d_1}{d_2 - b_2} > 0.$$

Notice now that stochastic process counterpart to the two parameter logistic growth equation has four parameters. Thus, there are an infinite number of parameter combinations in the stochastic model that correspond the determinisitic model! We see then that using the stochastic model to describe a specific system requires additional tuning.

The differences between the stochastic and deterministic models do not end there. It can be shown (we will not do this here) that for the logistic growth process, we have $\lim_{t\to\infty} p_0(t) = 1$, meaning that the population will eventually decay to zero. This is a very different prediction from the deterministic case for which the non-zero carrying capacity is the stable fixed point. We add though (again without proof) that if one were to measure the expected time to extinction, this increases rapidly with the carrying capacity of the system.

What we will analyse in detail is the mean and how that evolves with time. Based on the discussion above, we see that the transition probabilities are

$$p_{i+j,i}(\Delta t) = \text{Prob}\left\{X(t+\Delta t) = i+j | X(t) = i\right\} \tag{6}$$

$$= \begin{cases} \lambda_i \Delta t + o(\Delta t), & j = 1 \\ 1 - (\lambda_i + \mu_i)\Delta t + o(\Delta t), & j = 0 \\ \mu_i \Delta t + o(\Delta t), & j = -1 \\ o(\Delta t), & j \neq -1, 0, 1, \end{cases} \tag{7}$$

and, accordingly, the non-zero entries of the generator matrix will be

$$\begin{aligned} q_{ii} &= -(\lambda_i + \mu_i), \\ q_{i+1,i} &= \lambda_i, \\ q_{i-1,i} &= \mu_i. \end{aligned}$$

The precise expressions for λ_i and μ_i are provided in (5). As usual, we'll begin with the forward Kolmogorov equation. If we have $X(0) = N$, then the state probability is given by $p_i(t) = p_{iN}(t)$ and, therefore,

$$\frac{dp_i}{dt} = -(\lambda_i + \mu_i)p_i(t) + \lambda_{i-1}p_{i-1}(t) + \mu_{i+1}p_{i+1}(t). \tag{8}$$

To obtain the equation for the mean, we multiply (8) by i and after also summing over i, we have

$$\frac{dm}{dt} = -\sum_i i(\lambda_i + \mu_i)p_i(t) + \sum_i i\lambda_{i-1}p_{i-1}(t) + \sum_i i\mu_{i+1}p_{i+1}(t).$$

The sums on the right hand size can be written as

$$\begin{aligned} \sum_i i(\lambda_i + \mu_i)p_i(t) &= \sum_i i\left((b_1 + d_1)i + (b_2 + d_2)i^2\right)p_i(t) \\ \sum_i i\lambda_{i-1}p_{i-1}(t) &= \sum_i (i+1)\left(b_1 i + b_2 i^2\right)p_i(t) \\ \sum_i i\mu_{i+1}p_{i+1}(t) &= \sum_i (i-1)\left(d_1 i + d_2 i^2\right)p_i(t). \end{aligned}$$

Using these expressions in the equation for the mean, we have

$$\frac{dm}{dt} = \sum_i \left((b_1 - d_1)i - (d_2 - b_2)i^2\right)p_i(t),$$

and since $\sigma^2(t) = \sum_i i^2 p_i(t) - m^2(t)$, we arrive at

$$\frac{dm}{dt} = rm\left(1 - \frac{m}{K}\right) - r\frac{\sigma^2(t)}{K} \tag{9}$$

where $r = (b_1 - d_1)$ and $K = (b_1 - d_1)/(d_2 - b_2)$.

Examining (9), we notice a couple of things. The first is that the equation for the mean of the stochastic logistic process is not the logistic equation – it contains the additional term, $-r\sigma^2(t)/K$, that is proportional to the variance. This arises due to the nonlinear dependence of the birth and death rates on the population size.

A second property that we observe is that since $\sigma^2(t) \geq 0$, $m(t)$ will always be bounded from above by the solution to the logistic equation and, moreover, since $\lim_{t\to\infty} p_0(t) = 1$, we must also have $\lim_{t\to\infty} m(t) = 0$. This is qualitatively different to the logisitic equation which has $u^* = K$ as the stable fixed point. The caveat to keep in mind, however, is that it will likely take an exceptionally long time for the mean to reach 0.

0.3 Stochastic SIS model

Recall the SIS model,

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta}{N}SI + \gamma I,\\ \frac{dI}{dt} &= \frac{\beta}{N}SI - \gamma I,\end{aligned}$$

where S is the susceptible population, I is the infective population, and $N = S + I$ is a constant. We also have the recovery rate constant, γ, and the contact or transmission rate constant β.

Using $S = N - I$, we have then that

$$\frac{dI}{dt} = \frac{\beta}{N}I(N - I) - \gamma I,$$

which, if we recall, is equivalent to the logistic equation.

Fortunately, even though there are the notable differences that arise between the stochastic logistic process and the logistic equation, we are able to ascertain the birth and death rates for the stochastic process from the rates in the logistic equation. Namely,

$$\lambda_i = \begin{cases} \frac{\beta}{N}i(N-i) & i = 0, 1, 2, \ldots, N \\ 0 & i > N \end{cases}$$

and

$$\mu_i = \gamma i.$$

for $i = 0, 1, \ldots, N$. Before stating the transition probabilities, notice that we have limited ourselves to a finite state space, $i \in \{0, 1, \ldots, N\}$. This makes perfect sense given that our population size is fixed at N, so the largest the infective population can be is N.

Also notice that we have identified specific values of μ_i and λ_i even though the logistic equation allows for an infinite number of choices for the birth and death rates. Why is that? Well, for the SIS model we have two clearly distinct processes that are occurring – transmission and recovery. The rates in the stochastic process maintain this distinction, even though in the deterministic model they can be lumped to yield the standard logistic equation.

With the rates in hand, we have for the transition probabilities,

$$p_{i+j,i} = \begin{cases} \frac{\beta}{N} i(N-i) + o(\Delta t), & j = 1 \\ 1 - \left(\frac{\beta}{N} i(N-i) + \gamma i\right) + o(\Delta t), & j = 0 \\ \gamma i + o(\Delta t), & j = -1 \\ o(\Delta t), & j \neq -1, 0, 1 \end{cases}$$

As with any logistic growth process, the stochastic SIS model will have $\lim_{t\to\infty} p_0(t) = 1$, which is again different from our prediction using the deterministic SIS model where a finite number of infectives was a stable fixed point. This is welcomed news as it means the infective population will be reduced to zero, but it may take a rather long time. For example, if $\beta = 2$, $N = 100$, and $\gamma = 1$, the expected time that the infected population reaches zero will be on the order of 10^8 (this can be done using additional techniques not discussed in this module). If the rates are in units of 1/days, this means that it will take on average around 350000 years for the epidemic to end!

0.4 Quasistationary probability distribution

To put this number into context, humans have existed as a species for about 300000 years, so the fact the disease eventually goes away, doesn't give us much insight into what we should expect before it does. What we would like to do then is characterise the distribution prior to extinction with the assumption that the distribution will be more or less stationary during that time. This assumption makes sense given the long expected time to extinction, and also the fact that the stable fixed point of the logistic equation is non-zero. This distribution is known as the *quasistationary probability distribution.*

We see that the reason that we have extinction, albeit after a potentailly very long time, is that $X = 0$ is an absorber. We would like to remove this effect and insolate the $X = 0$ state by not permitting transitions to it. As we have,

$$\frac{dp_0}{dt} = \mu_1 p_1$$

(note that $\lambda_0 = 0$), we can remove the $X = 0$ state from the process by having $\mu_1 = 0$. Thus, to obtain the quasistationary probability distribution, we consider the slightly altered generator matrix,

$$\tilde{Q} = \begin{bmatrix} -\lambda_1 & \mu_2 & & & \\ \lambda_1 & -(\mu_2 + \lambda_2) & \mu_3 & & \\ & \lambda_2 & -(\mu_3 + \lambda_3) & \mu_4 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}.$$

which, you can see is the original generator matrix with the first column and first row removed. The quasistationary probability distribution is then given by $\tilde{Q}\tilde{\pi} = 0$ with $\sum_{i=1}^{\infty} \tilde{\pi} = 1$. Thus, the quasistationary distribution is given by

$$\tilde{\pi}_i = \frac{\lambda_1 \lambda_2 \cdots \lambda_{i-1}}{\mu_2 \mu_3 \cdots \mu_i} \tilde{\pi}_1$$

with the constraint that $\sum_{i=1}^{\infty} \tilde{\pi} = 1$.

Let's now consider the stochastic SIS model with $N = 100$, $\beta = 2$ and $\gamma = 1$. For the deterministic model the carrying capacity is $K = N(\beta - \gamma)/\beta$, which for these parameter values is $K = 50$. Some good questions to ask are: does the quasistationary probability distribution accurately characterise the distribution of states before extinction? How close is the mean of the quasistationary probability distribution to that of the full process, and how close are both to the carrying capacity of the deterministic system?

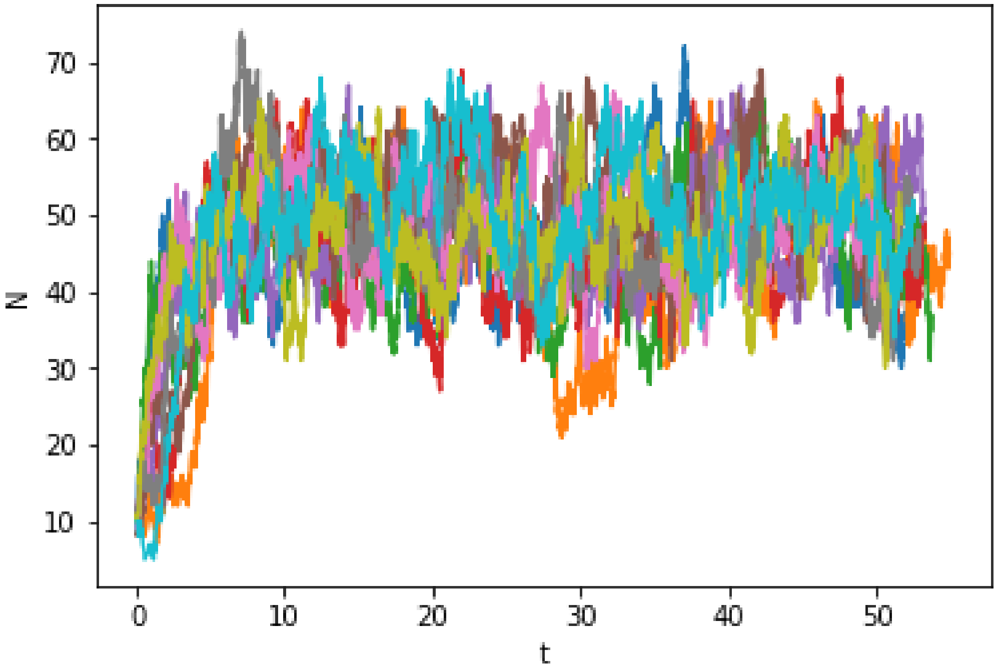
The cell below uses Gillespie's algorithm to generate sample trajectories for the stochastic SIS model. It computes the mean based on these trajectories and compares it to the mean given by the quasistationary probability distribution. We see that both yield values very close to the carrying capacity of the deterministic system, though slightly less in the case of the quasistationary probability distribution and always nearly less for the sample paths. This is to be expected since we know that the mean should be less than the solution to the logistic equation.

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

N = 100
beta = 2
gam = 1

nind = np.arange(4999);
nind2 = np.arange(100)+1;
nsamp = np.arange(10);
tgill = np.zeros(5000);
```

```
n = np.zeros(5000);

nmean = 0;
ntotsamp = 0;

for j in nsamp:
    tgill[0] = 0;
    n[0] = 10;
    for i in nind:
        lam = beta*n[i]*(N - n[i])/N
        mu = gam*n[i]
        tgill[i+1] = tgill[i] - np.log(np.random.rand(1,1))/((lam + mu))
        bord = np.random.rand(1,1);
        if bord < lam/(lam + mu):
            n[i+1] = n[i] + 1
        else:
            n[i+1] = n[i] - 1
        if tgill[i] > 10:
            nmean = nmean + n[i+1]
            ntotsamp = ntotsamp + 1

    plt.plot(tgill, n);

pivec = np.zeros(101);

pivec[0] = 1;
pisum = 0;

for j in nind2:
    lam = beta*j*(N - j)/N
    mu = gam*(j + 1)
    pivec[j] = lam*pivec[j-1]/mu
    pisum = pisum + pivec[j]

pivec[0] = 1 /(1 + pisum);
pivec[1:] = pivec[1:]*pivec[0];

qsdmean = np.sum(pivec[:100]*nind2);

plt.xlabel('t')
plt.ylabel('N');
print('mean from sample trajectories: ', nmean/ntotsamp)
print('mean from quasistationary distribution: ', qsdmean)
```

```
mean from sample trajectories:  49.24876782150843
mean from quasistationary distribution:  48.93044994826744
```



0.5 —- ALL MATERIAL BELOW THIS POINT WILL NOT BE EXAMINED! —-

0.6 Branching processes

The birth and death processes described above do not pay much attention to what individuals might do. The transition probabilities simply describe how the population goes from one size to another. For a branching process, the set up is slightly different.

What we are going to consider now is what is called a Markov, continuous-time branching process, $\{X(t), t \in [0, \infty)\}$, for random variable $X(t)$. Suppose that the time, τ, that a birth event occurs is exponentially distributed such that

$$G(t) = \text{Prob}\{\tau \leq t\} = 1 - e^{-\lambda t}.$$

When the event occurs, let's allow for the fact that there may be multiple offspring and let q_k be the probability that there are k members after the event. This may seem to be overkill thinking about human populations and birth events, but very appropriate for viruses, say, that take over cells that then release many more viruses at the birth event. The probability generating function for this is

$$f(z) = \sum_{k=0}^{\infty} q_k z^k.$$

This is known as the next generation probability generating function. Thus in our model at each event the parent can give rise to k offspring, each behaving independently and can give rise to subsequent generations based on the same probabilities, q_k.

Now, let's consider the total population size, which is given by the random variable $X(t)$. The name of the game will be to derive an equation for $\mathcal{P}(z, t)$, the probability generating function for X assuming that $X(0) = 1$.

Since at $t = 0$, $\mathcal{P}(z, 0) = z$, i.e. we have a single individual, then after a short time Δt later,

$$\begin{aligned}\mathcal{P}(z, \Delta t) &= z\text{Prob}\left\{\tau > \Delta t\right\} + f(z)\text{Prob}\left\{\tau \leq \Delta t\right\} + o(\Delta t)\\ &= ze^{-\lambda\Delta t} + f(z)(1 - e^{-\lambda\Delta t}) + o(\Delta t).\end{aligned}$$

As our aim is a differential equation, we'll now consider the quantity $\mathcal{P}(z, t + \Delta t) - \mathcal{P}(z, t)$. Before doing so, we'll utilise a key fact that is left without proof, namely,

$$\mathcal{P}(z, t + \Delta t) = \mathcal{P}(\mathcal{P}(z, t), \Delta t),$$

again, for sufficiently small Δt. This tells us that the probability generating function can be evolved an infinitesimal amount of time through a composition with itself. Using this, we have

$$\begin{aligned}\mathcal{P}(z, t + \Delta t) - \mathcal{P}(z, t) &= \mathcal{P}(\mathcal{P}(z, t), \Delta t) - \mathcal{P}(z, t)\\ &= \mathcal{P}(z, t)e^{-\lambda\Delta t} + f\left(\mathcal{P}(z, t)\right)(1 - e^{-\lambda\Delta t}) - \mathcal{P}(z, t) + o(\Delta t).\\ &= \left[-\mathcal{P}(z, t) + f\left(\mathcal{P}(z, t)\right)\right](1 - e^{-\lambda\Delta t}) + o(\Delta t).\end{aligned}$$

Dividing both sides by Δt and taking the limit $\Delta t \to 0$, gives

$$\frac{d\mathcal{P}}{dt} = -\lambda\left[\mathcal{P}(z, t) - f\left(\mathcal{P}(z, t)\right)\right],$$

where we have introduced the total derivative as the right hand side depends on z parametrically. This equation is known as the backward Kolmogorov differential equation. Notice that the fixed point satisfies $f(z) = z$.

Before examining an example, it is important to highlight a key fact regarding continuous-time branching processes. The mean number of births is given by $m = \sum_{k=0}^{\infty} kq_k = f'(1)$. The asymptotic behaviour of the process depends on whether $m \leq 1$ or $m > 1$. Namely, if $m \leq 1$ then $\lim_{t\to\infty} \text{Prob}\left\{X(t) = 0\right\} = \lim_{t\to\infty} p_0(t) = 1$ and if $m > 1$ then there exists $c < 1$ satisfying $f(c) = c$, and $\lim_{t\to\infty} \text{Prob}\left\{X(t) = 0\right\} = \lim_{t\to\infty} p_0(t) = c$. This tells us that if the mean number of births at each event is too low, the population will certainly vanish.

0.6.1 Simple birth and death as a branching process

Let's take the case where $G(t) = 1 - e^{-(\lambda+\mu)t}$ and

$$f(z) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} z^2.$$

This means that probability of zero members after the event is $\mu/(\lambda+\mu)$, while the probability of two members is $\lambda/(\lambda+\mu)$. Thus, the population will decrease by one with probability $\mu/(\lambda+\mu)$ or increase by one with probability $\lambda/(\lambda+\mu)$.

With this expression for $f(z)$, the differential equation for $\mathcal{P}(z,t)$ is

$$\frac{d\mathcal{P}}{dt} = -(\lambda+\mu)\mathcal{P} + \mu + \lambda\mathcal{P}^2.$$

with the initial condition $\mathcal{P}(z,0) = z$. This is a separable equation, which, without going through the details has solution,

$$\mathcal{P}(z,t) = \frac{e^{(\mu-\lambda)t}(\lambda z - \mu) - \mu(z-1)}{e^{(\mu-\lambda)t}(\lambda z - \mu) - \lambda(z-1)}, \lambda \neq \mu.$$

This coincides with our expression for the generating function for the simple birth and death process in the case of $N = 1$. Then, we have that

$$p_0(\infty) = \left\{ \begin{array}{ll} 1, & \text{if} \lambda \leq \mu \\ \frac{\mu}{\lambda}, & \text{if} \lambda > \mu. \end{array} \right. ,$$

which we see also corresponds to the solution to,

$$f(z) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} z^2 = z,$$

for $\lambda > \mu$.

0.6.2 Multitype branching processes

A nice aspect of branching processes is that they readily extend to cases where the individuals can be of multiple type. Here, the events that take place allow the individuals to give rise to different types with a certain probability. Specifically, we will consider k individuals and denote $X_i(t)$ as the number of individuals of type i for $i = 1, \ldots, k$, at time t. Additionally, we will have the next generation probability generating function for type i, $f_i(z_1, z_2, \ldots, z_k)$, and the generating function for type i, $\mathcal{P}_i(z_1, z_2, \ldots, z_k, t)$. The $f_i(z_1, z_2, \ldots, z_k)$ will tell us how many of each type k will we might expect to arise from a single type i. The generating functions $\mathcal{P}_i(z_1, z_2, \ldots, z_k, t)$ correspond to processes that begin with $X_i(0) = 1$ and $X_j(0) = 0$ for $j \neq i$.

If again $G(t) = 1 - e^{-\lambda t}$, then the generating functions are given by the differential equations,

$$\frac{d\mathcal{P}_i}{dt} = -\lambda \left[\mathcal{P}_i - f_i(\mathcal{P}_1, \ldots, \mathcal{P}_k)\right], \text{ for } i = 1, \ldots, k.$$

As is usually the case, the complexity is all in the notation that one must introduce when one has multiple types or species. The simplicity of this is best illustrated by an example, and a nice one at that.

Drug resistant cancer cells Let's consider a tumour that consists of two types of cancer cells – ones that are sensitive to a particular drug, and others that are resistant to the treatment. Let's call the sensitive cells type 1 and the resistant, type 2. At cell division, a type 1 cell will give rise to one type 1 cell and either another type 1 with probability $1 - p$, or a mutation occurs and the daughter cell is instead type 2. For type 2 cells, the daughter cell will be of type 2. Based on this, the next generation generating functions will be

$$\begin{aligned} f_1(z_1, z_2) &= (1 - p)z_1^2 + pz_1 z_2 \\ f_2(z_1, z_2) &= z_2^2. \end{aligned}$$

As a result, we have then,

$$\begin{aligned} \frac{d\mathcal{P}_1}{dt} &= -\lambda \left[\mathcal{P}_1 - (1 - p)\mathcal{P}_1^2 - p\mathcal{P}_1\mathcal{P}_2\right], \\ \frac{d\mathcal{P}_2}{dt} &= -\lambda \left[\mathcal{P}_2 - \mathcal{P}_2^2\right] \end{aligned}$$

with initial conditions $\mathcal{P}_1(z_1, z_2, 0) = z_1$ and $\mathcal{P}_2(z_1, z_2, 0) = z_2$. It is now just a matter of solving these differential equations. The differential equation for $\mathcal{P}_2$ is the logistic equation with $r = -\lambda$ and $K = 1$. The solution (as we've seen before) is given by

$$\mathcal{P}_2 = \frac{z_2}{z_2 + (1 - z_2)e^{\lambda t}}.$$

Upon substituting this into the differential equation for $\mathcal{P}_1$ and solving, one finds that

$$\mathcal{P}_1 = \frac{z_1 e^{-\lambda t}\left[z_2 e^{-\lambda t} + 1 - z_2\right]^{-p}}{1 + z_1\left(\left[z_2 e^{-\lambda t} + 1 - z_2\right]^{1-p} - 1\right) z_2^{-1}}.$$

In order to successfully treat the tumour, one would like to ensure that when the tumour is found, there are no resistant cells. Taking the case where $X_1(0) = 1$ and $X_2(0) = 0$, the probability that there won't be any resistant cells at time t is,

$$\lim_{z_1 \to 1} \lim_{z_2 \to 0} \mathcal{P}_1(z_1, z_2, t) = \frac{1}{1 - p + pe^{\lambda t}}.$$

Given the nature of the process, it's perhaps not surprising that the probability that there are no resistant cells decreases monotonically with time and, in fact, as $t \to \infty$ at least one cell will be resistant. Thus, it's best to find the tumour as early as possible.

0.7 Multivariate processes

In our discussion of birth and death processes above, we considered stochastic processes that describe a single random variable. These are known as univariate processes. We saw earlier in the module that in order to model biological phenomena involving interacting populations or multiple chemical species, we had to extend our models to multidimensional systems of ODEs. To describe similar phenomena using stochastic processes, we need to extend to *multivariate* processes.

Let's consider the case of a continuous-time bivariate Markov process, where we have two random variables $X(t)$ and $Y(t)$ for $t \geq 0$ with $X(t)$ and $Y(t) \in \{0, 1, 2, \dots\}$. We'll be interested in computing the probability that at time t, $X(t) = m$ and $Y(t) = n$, that is

$$p_{(m,n)}(t) = \text{Prob}\left\{X(t) = m, Y(t) = n\right\}.$$

This quantity is known as the joint probability mass function. The transition probability for the process will be

$$p_{(m,n),(i,j)}(\Delta t) = \text{Prob}\left\{X(t + \Delta t) = m, Y(t + \Delta t) = n | X(t) = i, Y(t) = j\right\}$$

and, since we assume that the process is homogeneous, the transition probabilities will only depend on Δt, not t. With this in mind, we express the transition probabilities as

$$p_{(i+k,j+l),(i,j)}(\Delta t) = \begin{cases} h_{kl}(i,j)\Delta t + o(\Delta t), & (k,l) \in S \\ 1 - \Delta t \sum_{(k,l)\in S} h_{ij}(i,j) + o(\Delta t), & (k,l) \notin S \end{cases}$$

where S is a finite subset $\mathbb{Z} \times \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers. Take, for example, $S = \{(1,0), (0,1), (-1,0), (0,-1)\}$, with $h_{10}(i,j) = \lambda_1 i$, $h_{01}(i,j) = \lambda_2 j$, $h_{-1,0}(i,j) = \mu_1 i$, and $h_{0,-1}(i,j) = \mu_2 j$, then the transition probabilities are

$$p_{(i+k,j+l),(i,j)}(\Delta t) = \begin{cases} \lambda_1 i \Delta t + o(\Delta t), & (k,l) = (1,0) \\ \lambda_2 j \Delta t + o(\Delta t), & (k,l) = (0,1) \\ \mu_1 i \Delta t + o(\Delta t), & (k,l) = (-1,0) \\ \mu_2 j \Delta t + o(\Delta t), & (k,l) = (0,-1) \\ 1 - \Delta t \left((\lambda_1 + \mu_1)i + (\lambda_2 + \mu_2)j\right) + o(\Delta t), (k,l) = (0,0) \end{cases}$$

As we had done in the univariate case, we can use the transition probabilities to derive the forward Kolmogorov differential equation for the bivariate process. Considering again the general form of the transition probabilities, for the joint probability mass function we'll have,

$$p_{(m,n)}(t+\Delta t) = \sum_{(k,l)\in S} p_{(m-k,n-l)}(t)h_{kl}(m-k,n-l)\Delta t + p_{(m,n)}(t)\left(1-\Delta t\sum_{(k,l)\in S} h_{kl}(m,n)\right).$$

Subtracting both sides by $p_{(m,n)}(t)$, dividing through by Δt and then taking the limit $t \to 0$, we arrive at the forward Kolmogorov differential equation

$$\frac{dp_{(m,n)}}{dt} = -p_{(m,n)}(t)\sum_{(k,l)\in S} h_{kl}(m,n) + \sum_{(k,l)\in S} p_{(m-k,n-l)}(t)h_{kl}(m-k,n-l).$$

For our specific example introduced above, the forward Kolmogorov differential equation will be

$$\begin{aligned}\frac{dp_{(m,n)}}{dt} &= -p_{(m,n)}(t)\left((\lambda_1+\mu_1)m + (\lambda_2+\mu_2)n\right)\\ &\quad + \lambda_1(m-1)p_{(m-1,n)}(t) + \lambda_2(n-1)p_{(m,n-1)}(t)\\ &\quad + \mu_1(m+1)p_{(m+1,n)}(t) + \mu_2(n+1)p_{(m,n+1)}(t).\end{aligned}$$

It is also possible to construct the generating function,

$$\mathcal{P}(w,z,t) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} p_{(m,n)}(t)w^m z^n$$

for a bivariate process. The marginal probability distributions are given by $\sum_{n=0}^{\infty} p_{(m,n)}(t)$ and $\sum_{m=0}^{\infty} p_{(m,n)}(t)$, for $X(t)$ and $Y(t)$, respectively, and from which we can compute the means

$$\begin{aligned}\mu_X &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} mp_{(m,n)}(t),\\ \mu_Y &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} np_{(m,n)}(t),\end{aligned}$$

and variances,

$$\begin{aligned}\sigma_X^2 &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} m^2 p_{(m,n)}(t) - \mu_X^2,\\ \sigma_Y^2 &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} n^2 p_{(m,n)}(t) - \mu_Y^2.\end{aligned}$$

We see that many of the concepts and techniques from univariate processes carry over to multivariate processes, but with the additional technical complications of indexing and more dependent variable appearing in $\mathcal{P}$. Unfortunately, these technical complications do come with the baggage that they render our analytical approaches somewhat useless, leaving computational methods as the way forward.

0.8 Stochastic enzyme dynamics

In the first weeks, we studied the Michaelis-Menten system,

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} SE \xrightarrow{k_2} P + E \tag{10}$$

for enzyme dynamics. After applying mass action and reducing the system, we arrived at,

$$\frac{dc}{dt} = k_1 s(e_0 - c) - (k_{-1} + k_2)c, \tag{11}$$

$$\frac{ds}{dt} = -k_1 s(e_0 - c) + k_{-1}c \tag{12}$$

for c, the concentration of the complex SE, and s, the concentration of the substrate, S. Recall also that we invoked the quasi-steady approximation where $dc/dt \approx 0$ and therefore,

$$c = \frac{k_1 e_0 s}{k_{-1} + k_2 + k_1 s}$$

and

$$\frac{ds}{dt} = -\frac{k_{max} s}{k_m + s}$$

where $k_{max} = k_2 e_0$ and $k_m = (k_2 + k_{-1})/k_1$.

Based on this model, we now would like to develop the corresponding bivariate birth and death process for the enzyme reaction. The first thing to notice is that we need to be careful about the dimensions of the quantities – our stochastic process keeps track of the number of things, but these equation are for concentrations. That means for our stochastic process, the rates will have to have dimensions of [molecules]/[time] while in our deterministic models the rates have dimension [concentration]/[time]. How do we account for this? Well, we see that parameters k_{-1} and k_2 have dimensions of inverse time. This means that, if we replace concentrations in the deterministic model by numbers of molecules when we write down the transition probabilities, everything works out and we are in business. The coefficient k_1, however, has dimensions of 1/[concentration × time]. This means that we would need to convert k_1 into another quantity, $\tilde{k}_1$ which has dimensions of 1/[molecules × time]. I'll spare us the high school chemistry lesson, but this conversion will involve things like Avogadro's number and all of that business. For our purposes, we'll recognise that it needs to be converted to a quantity with the correct dimensions, which we'll call $\tilde{k}_1$. The same goes for the concentration e_0 which we convert to the number of molecules $\tilde{e}_0$.

We are almost there. The next thing to recognise is what value the system should transition to given n molecules of S and b molecules of SE. We see that the term $k_1 s(e_0 - c)$ appears in the equation for c, while $-k_1 s(e_0 - c)$ appears in that for s. Thus, the corresponding transition probability will be $p_{n-1,b+1}(\Delta t)$. Likewise, the transition probability corresponding to the term $k_{-1}c$ in the deterministic equations will be $p_{n+1,b-1}(\Delta t)$. Finally, we have $k_2 c$ which only decreases the concentration of c and therefore, the corresponding transition probability will be $p_{n,b-1}(\Delta t)$.

Putting these facts together, we see that the transition probabilities are

$$p_{(n+i,b+j),(n,b)}(\Delta t) = \begin{cases} \tilde{k}_1 n(\tilde{e}_0 - b)\Delta t + o(\Delta t), & (i,j) = (-1,1) \\ k_{-1} b\Delta t + o(\Delta t), & (i,j) = (1,-1) \\ k_2 b\Delta t + o(\Delta t), & (i,j) = (0,-1) \\ 1 - \left[\tilde{k}_1 n(\tilde{e}_0 - b) + k_{-1}b + k_2 b\right]\Delta t + o(\Delta t), & (i,j) = (0,0) \\ o(\Delta t), & \text{otherwise} \end{cases}$$

Applying the ideas above that we observed for a general multivariate process, we see that the forward Kolmogorov differential equation is

$$\begin{aligned} \frac{dp_{(n,b)}}{dt} &= \tilde{k}_1(n+1)(\tilde{e}_0 - (b-1))p_{(n+1,b-1)} + k_{-1}(b+1)p_{(n-1,b+1)} \\ &+ k_2(b+1)p_{(n,b+1)} - \left[\tilde{k}_1 n(\tilde{e}_0 - b) + k_{-1}b + k_2 b\right] p_{(n,b)}. \end{aligned}$$

We can see how complicated things can become once we enter the land of multivariate processes. It's useful then to consider the univariate stochastic process based on the quasisteady equations. Here, the transition probabilities will be

$$p_{n+i,n}(\Delta t) = \begin{cases} \tilde{k}_{max} n\Delta t/(\tilde{k}_m + n) + o(\Delta t), & i = -1 \\ 1 - \tilde{k}_{max} n\Delta t/(\tilde{k}_m + n) + o(\Delta t), & i = 0 \\ o(\Delta t), & i \le -2 \\ 0, & i > 0 \end{cases}$$

where $\tilde{k}_{max} = k_2\tilde{e}_0$ and $\tilde{k}_m = (k_{-1} + k_2)/\tilde{k}_1$. Thus, this is a death process with death rate $\mu_n = \tilde{k}_{max} n/(\tilde{k}_m + n)$. Recall from our studies of the deterministic model that as a result of the quasisteady approximation, we have $dp/dt = -ds/dt$, where p is the concentration of the product P. As a result, the death process for the substrate can be flipped around and be viewed as a birth process with birth rate $\lambda_n = \tilde{k}_{max} n/(\tilde{k}_m + n)$ for the product.

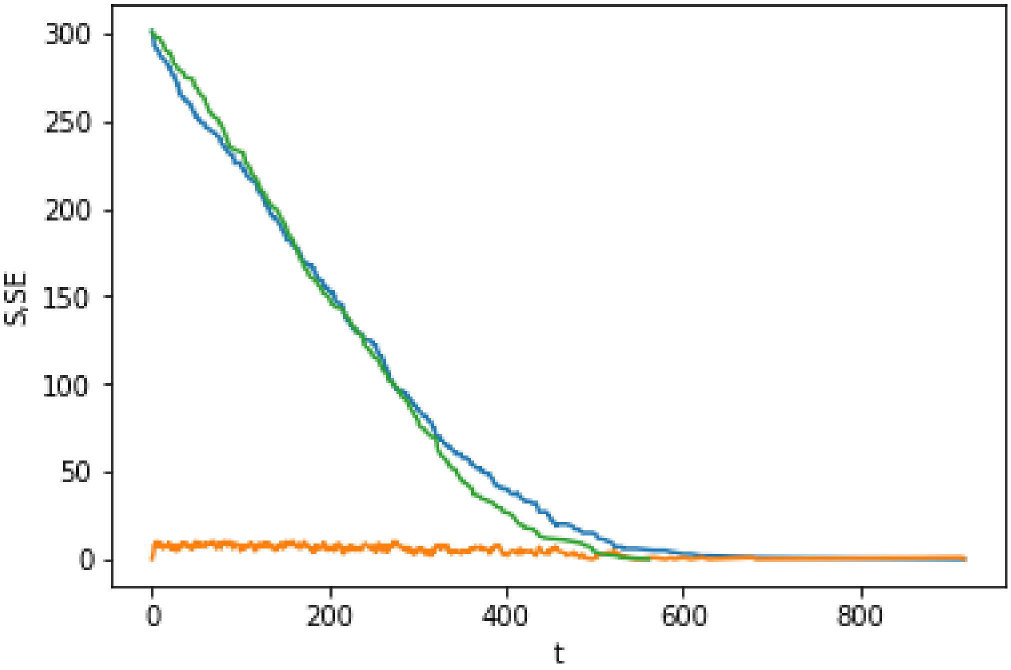
The cell below compares results from the multivariate stochastic process with those from the model generated using the quasisteady approximation. As with the deterministic system, the approximation works well in the limit where $\tilde{e}_0$ is small. Take $\tilde{e}_0 = 12$ in the cell below. Here, $S(t)$ –

the number of substrate molecules – given by both stochastic processes are very similar, while the number of complex molecules $SE(t)$ fluctuate about a nearly constant value. The approximation degrades as $\tilde{e}_0$ increases.

[5]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

k1til = 0.00166
km1 = 0.0001
k2 = 0.1
einf = 10
pinf = 301

kmax = k2*einf
km = (km1 + k2)/k1til

nind = np.arange(999);
nsamp = np.arange(1);
tgill = np.zeros(1000);
n = np.zeros(1000);
b = np.zeros(1000);
nqs = np.zeros(1000);
tqs = np.zeros(1000);

for j in nsamp:
    tgill[0] = 0;
    n[0] = pinf;
    b[0] = 0;
    nqs[0] = pinf;
    for i in nind:
        r1 = k1til*n[i]*(einf - b[i])
        r2 = km1*b[i]
        r3 = k2*b[i]
        rtot = r1 + r2 + r3;
        tgill[i+1] = tgill[i] - np.log(np.random.rand(1,1))/(rtot)
        bord = np.random.rand(1,1);
        b[i+1] = b[i]
        n[i+1] = n[i]

        rqs = kmax*nqs[i]/(km + nqs[i]);
        tqs[i+1] = tqs[i] - np.log(np.random.rand(1,1))/(rqs)
        nqs[i+1] = nqs[i] - 1
        if bord < r1/rtot:
            n[i+1] = n[i] - 1
            b[i+1] = b[i] + 1
        elif r1/rtot < bord and bord < (r1+r2)/rtot:
            n[i+1] = n[i] + 1
            b[i+1] = b[i] - 1
        else:
            b[i+1] = b[i] - 1

    plt.figure(1)
    plt.plot(tgill,n);
    plt.plot(tgill,b);
    plt.plot(tqs,nqs);
    plt.xlabel("t");
    plt.ylabel("S,SE");
```

```
/Users/ekeaveny/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:40:
RuntimeWarning: divide by zero encountered in true_divide
/Users/ekeaveny/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:34:
RuntimeWarning: divide by zero encountered in true_divide
/Users/ekeaveny/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:42:
RuntimeWarning: invalid value encountered in double_scalars
/Users/ekeaveny/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:45:
RuntimeWarning: invalid value encountered in double_scalars
```



0.9 Stochastic predator-prey dynamics

As a final example of a familiar model recast as a multivariate process, we consider the Lotka-Volterra system, which you may recall we expressed as

$$\begin{aligned}\frac{dN}{dt} &= aN - bNP,\\ \frac{dP}{dt} &= -dP + cNP,\end{aligned}$$

where N is the number of prey and P is the number of predators. This system emitted two fixed points, one at the origin which was unstable, and the other which we found to be a neutrally stable centre. The system produced oscillatory solutions (closed orbits in phase phase). The observed orbit depended on the initial conditions of the system.

Based on the terms in the deterministic system, we can find the transition probabilities. We see from the equation for the prey that the birth rate will be $\lambda_N = an$ (we will use n as the number of prey and r as the number of predators) and death rate $\mu_N = bnr$. For the predator population we have birth rate $\lambda_P = cnr$ and death rate $\mu_P = dr$. Based on this, we see that the transition probabilities are

$$p_{(n+i,r+j),(n,r)}(\Delta t) = \begin{cases} an\Delta t + o(\Delta t), & (i,j) = (1,0)\\ bnr\Delta t + o(\Delta t), & (i,j) = (-1,0)\\ cnr\Delta t + o(\Delta t), & (i,j) = (0,1)\\ dr\Delta t + o(\Delta t), & (i,j) = (0,-1)\\ 1 - [an + bnr + cnr + dr]\,\Delta t + o(\Delta t), & (i,j) = (0,0)\\ o(\Delta t), & \text{otherwise}\end{cases}$$

From these transition probabilities, we see that the forward Kolmogorov equation is then,

$$\begin{aligned}\frac{dp_{(n,r)}}{dt} &= a(n-1)p_{(n-1,r)} + b(n+1)rp_{(n+1,r)} + c(r-1)np_{(n,r-1)}\\ &\quad + d(r+1)p_{(n,r+1)} - (an + bnr + cnr + dr)\,p_{(n,r)}. \end{aligned} \tag{13}$$

It is instructive (though perhaps not necessarilly useful) to compute the differential equations for the mean values of the prey, $m_N = \sum_{(n,r)} np_{(n,r)}$, and predator, $m_P = \sum_{(n,r)} rp_{(n,r)}$, populations. Let's first do this for the prey population by multiplying (13) by n and summing over both n and r. In doing so, we find the following sums arise, which upon further manipulation become,

$$\begin{aligned}
\sum_{(n,r)} n\frac{dp_{(n,r)}}{dt} &= \frac{dm_N}{dt} \\
\sum_{(n,r)} an(n-1)p_{(n-1,r)} &= \sum_{(n,r)} an(n+1)p_{(n,r)} \\
\sum_{(n,r)} bn(n+1)rp_{(n+1,r)} &= \sum_{(n,r)} b(n-1)nrp_{(n,r)} \\
\sum_{(n,r)} cn^2(r-1)p_{(n,r-1)} &= \sum_{(n,r)} cn^2rp_{(n,r)} \\
\sum_{(n,r)} dn(r+1)p_{(n,r+1)} &= \sum_{(n,r)} dnrp_{(n,r)}.
\end{aligned}$$

By combining these with the sum obtained from the final term, we arrive at the relatively (and deceptively) simple equation

$$\frac{dm_N}{dt} = am_N - bm_{NP}$$

where $m_{NP} = \sum_{(n,r)} nrp_{(n,r)}$. For m_P, following a similar set of steps, we have

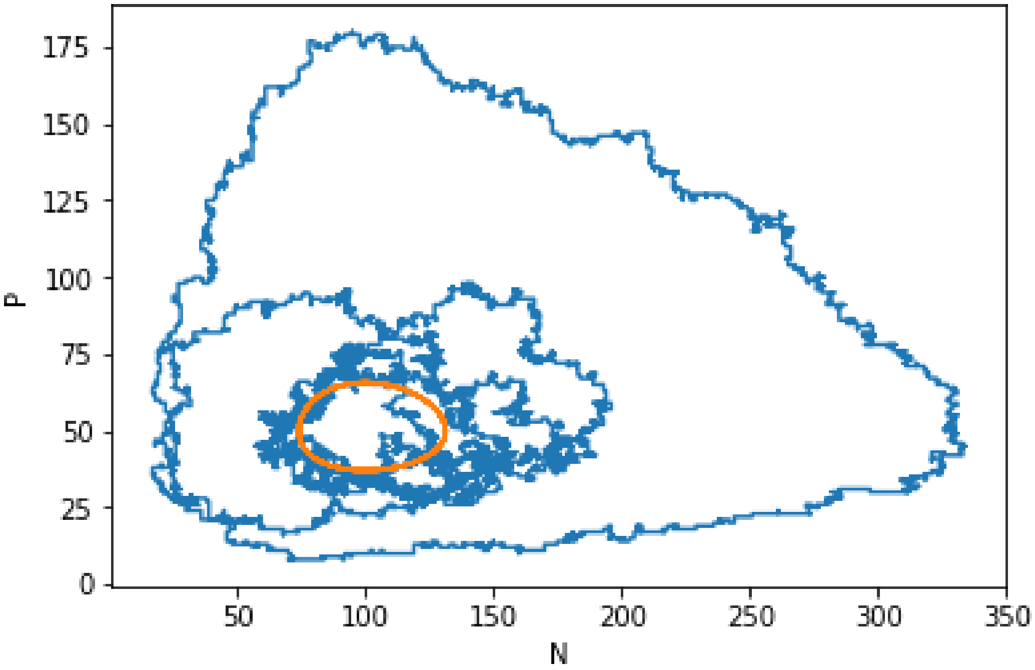
$$\frac{dm_P}{dt} = -dm_P + cm_{NP}.$$

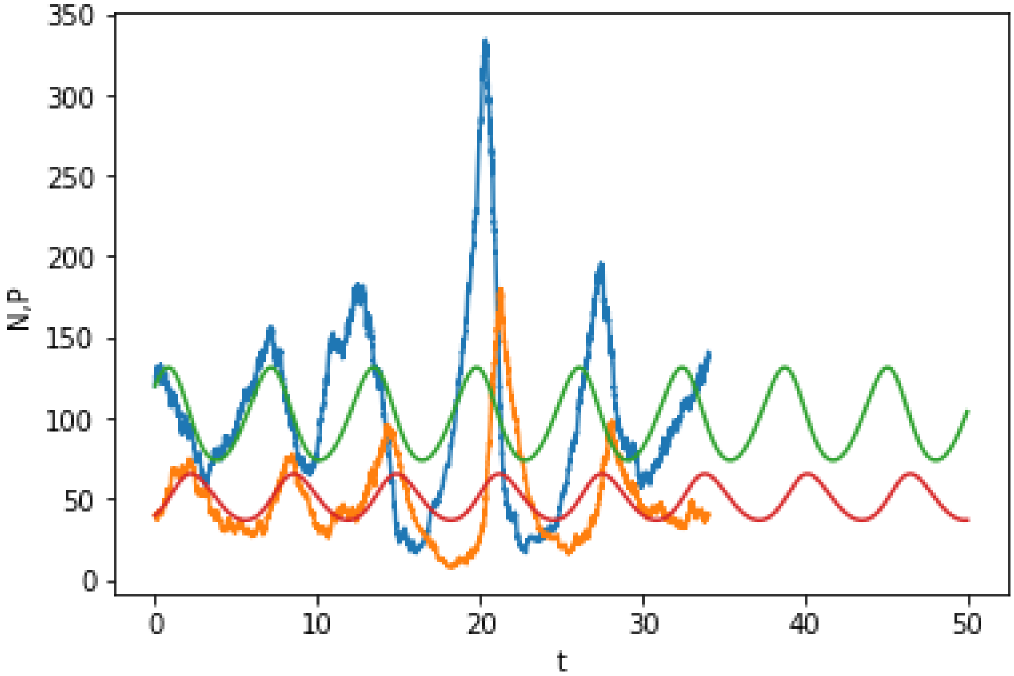
While these equations may seem relatively simple, there is a problem – we can't solve them as we don't know m_{NP}. If we go ahead and find the differential equation for m_{NP}, we would see that terms in this equation would also be unknown. This is known as a closure problem and is frequently seen with nonlinear problems like this one. In fact, we already saw this occur in the univariate logistic growth process. In practice, approximations that specify the unknown variables are usually invoked to close the system. One common approximation for m_{NP} would be $m_{NP} \approx m_N m_P$. In statistical terms, this means that the predator and prey populations are taken to be independent random variables, which we know is not true (otherwise the system would be closed on its own). We see, however, that this would produce the deterministic model!

Again, for these more complicated processes, numerical methods provide a path forward. The cell below generates and plots a sample path of the stochastic process, as well as the solution to the deterministic system. The top image shows the sample path and ODE solution in phase space, while the lower plot shows the predator and prey population sizes as a function of time. We see that the stochastic solution is able to experience large oscillations in the populations. This is likely due to the solution being able to hop from one closed orbit to another.

[4]:

```python
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.integrate import odeint
from scipy import linalg

a = 1
b = 0.02
c = 0.01
d = 1

def du_dt(u, t):
    return [a*u[0] - b*u[0]*u[1], -d*u[1] + c*u[0]*u[1]]

ts = np.linspace(0, 50, 1000)
u0 = [120, 40]
us = odeint(du_dt, u0, ts)
pop1 = us[:,0]
pop2 = us[:,1]

nind = np.arange(9999);
nsamp = np.arange(1);
tgill = np.zeros(10000);
n = np.zeros(10000);
r = np.zeros(10000);

for j in nsamp:
    tgill[0] = 0;
    n[0] = 120;
    r[0] = 40;
    for i in nind:
        lam1 = a*n[i];
        lam2 = c*n[i]*r[i];
        mu1 = b*n[i]*r[i];
        mu2 = d*r[i];
        lammutot = lam1 + lam2 + mu1 + mu2;
        tgill[i+1] = tgill[i] - np.log(np.random.rand(1,1))/(lammutot)
        bord = np.random.rand(1,1);
        r[i+1] = r[i]
        n[i+1] = n[i]
        if bord < lam1/lammutot:
            n[i+1] = n[i] + 1
        elif lam1/lammutot < bord and bord < (lam1+lam2)/lammutot:
            r[i+1] = r[i] + 1
        elif (lam1+lam2)/lammutot < bord and bord < (lam1+lam2 + mu1)/lammutot:
            n[i+1] = n[i] - 1
        else:
            r[i+1] = r[i] - 1

    plt.figure(0)
    plt.plot(n,r);
    plt.plot(pop1,pop2)
    plt.xlabel("N");
    plt.ylabel("P");

    plt.figure(1)
    plt.plot(tgill,n);
    plt.plot(tgill,r);
    plt.plot(ts,pop1);
    plt.plot(ts,pop2);
    plt.xlabel("t");
    plt.ylabel("N,P");
```




350
300
250
200
150
100
50
0
N,P
0
10
20
30
40
50
t

[ ]: