MA32009

Philip Murray

2023-12-22

Table of contents

Preface

Welcome to MA32009 (Mathematical Biology I).

My name is Philip Murray and I am the module lead.

Part I

Discrete time

1 Single species population dynamics in discrete time

Let us firstly consider models of a single population defined at discrete times. Discrete (rather than continuous) time is used in population models where, for example, there are strong annual trends in the underlying animal behaviour (e.g. fish eggs hatching at a fixed time in the year), there are not overlapping generations and the population size at the next time step can be written as a function of the current population size.

1.1 A general model for a single population in discrete time

Let t be an independent variable representing time and N_t be a dependent variable representing the size of the population of a given species at time t.

Consider the first order difference equation

$$N_{t+1} = N_t f(N_t) = H(N_t), \tag{1.1}$$

where $f(N_t)$ is a function that defines the per capita growth rate. The function $H(N_t)$ describes the total (net) growth rate.

Given some initial condition N_0 defined such that

$$N_{t=0} = N_0,$$

the goal of this section is to develop a range of techniques that allow us to analyse the behaviour of Equation 1.1 for a given H.

Finally, note that many of you were introduced to difference equations in the module MA21003 Discrete Mathematics. It is recommended that you revisit the section on Discrete Difference Equations.

1.2 The Malthusian model

Perhaps the simplest form for equation Equation 1.1 is when the net per capita growth rate is a constant.

Let b be the per capita growth rate and d the per capita death rate with b and d both positive real constants.

Hence, a population of size N_t at time t will grow by

$$bN_t$$

over the course of the next iteration and decay by

$$dN_t.$$

Therefore the population size at time $t+1$ is

$$N_{t+1} = N_t + bN_t - dN_t = rN_t, \tag{1.2}$$

where the parameter $r = 1 + b - d$ is known as the net growth rate.

Given the initial condition N_0, equation Equation 1.2 can be solved recursively as follows:

$$N_1 = rN_0, N_2 = r(N_1) = r^2N_0, N_3 = rN_2 = r^3N_0, ..., N_t = r^tN_0.$$

Hence for the case of Malthusian growth, the size of the population can be explicitly calculated for all t.

In Figure 1.1 a numerical solution of the model is computed for different values of the parameter r.

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

T=10

t = np.arange(0, T, 1)
N = np.zeros_like(t,dtype=float)
N_2 = np.zeros_like(t,dtype=float)

N_0=3.0
r=0.5
r_2=1.5

def rhs(x,r):
```

```
  f=r*x
  return f

def SolveSingleDiff(t,rhs,N_0,r):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r)
  return N

N=SolveSingleDiff(t,rhs,N_0,r)
N_2=SolveSingleDiff(t,rhs,N_0,r_2)


fig, ax = plt.subplots(1,2)
ax[0].plot(t, N)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
ax[1].plot(t, N_2)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
plt.show()
```

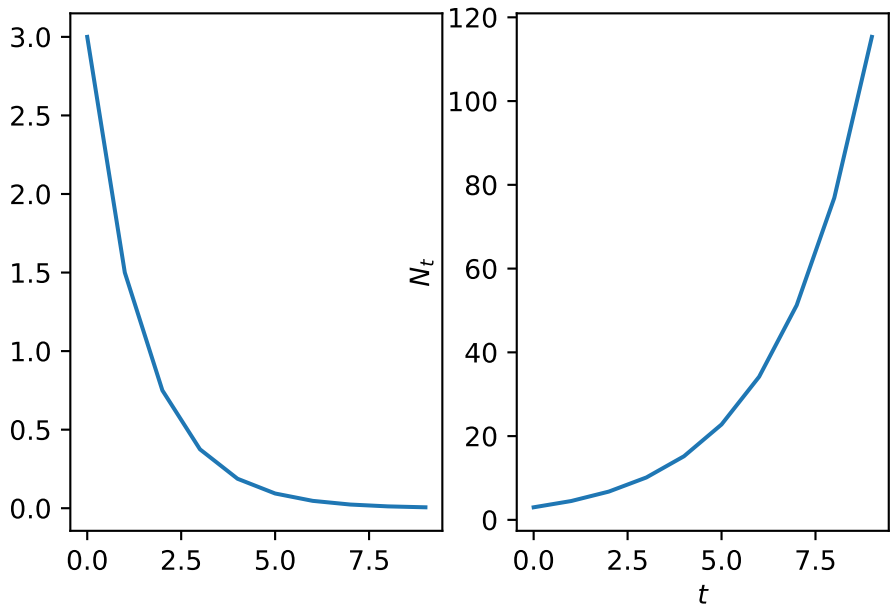

Figure 1.1: A plot of the Malthusian model solution when (a) r=0.5 and (b) r=1.5.

Can you describe the behaviour of the Malthusian model? How does it depend on the value of the parameter r?

Qualitative analyses allows us to categorise solution behaviours into different cases. For example, note that the behaviour of the population at large t is governed by the magnitude of the parameter r.

As $t \to \infty$

$$N \to \begin{cases} \infty & r > 1 \\ N_0 & r = 1 \\ 0 & r < 1. \end{cases}$$

In Figure 1.1 solutions in each of the (nontrivial) parameter regimes are plotted. When $r < 1$ the population tends to zero and when $r > 1$ the population explodes.

Whilst the simplicity of the Malthusian model allows us to calculate explicit solutions that yield insight into how the processes of birth and death combine to give either net growth or net reduction in population size, an obvious flaw with this model is that it displays unbounded growth for $r > 1$. In most biological systems, other effects, such as competition for resources or predation, will tend to limit a population's size. In single species population models, these effects are accounted for phenomenologically by introducing nonlinear terms into the function f.

1.3 A motivating example of a nonlinear model - The Ricker model

Many models of population dynamics in discrete time were initially developed to study fisheries. Well-studied examples include the Beverton Holt model

$$N_{t+1} = \frac{rN_t}{1 + \frac{N_t}{K}},$$

the Hassell model

$$N_{t+1} = \frac{rN_t}{(1 + \frac{N_t}{K})^b},$$

and the Ricker model

$$N_{t+1} = N_t e^{r(1-\frac{N_t}{K})}.$$

where r, K and b are positive parameters. Below we consider the Ricker model which was initially used to study Canadian sockeye salmon population dynamics.

1.3.1 Model development

One way to capture a decreased growth rate at high densities, is to assume that the per capita growth rate is an exponentially decreasing function of population density, i.e.

$$f(N_t) = e^{r(1-\frac{N_t}{K})},$$

where r is a growth rate and K a carrying capacity (both positive constants). Hence the governing equation is

$$N_{t+1} = N_t e^{r(1-\frac{N_t}{K})}. \tag{1.3}$$

Note that as the population size gets large ($N_t \gg K$), the growth rate tends to zero but that for small populations $N_t \ll K$ the growth rate is approximately constant. This is the Ricker model.

Note that the model is nonlinear and does not have an explicit solution.

1.3.2 Computational solutions

After directly computing numerical solutions of the Ricker model for different values of the parameter r, the data in (**plotricker?**) were obtained.

Note that model behaviour changes drastically as the parameter r varies. For instance,

- when $r = 0.5$ (Figure 1.2 (a)), the solution monotonically approaches the value 2, *
- *
  when $r = 1.5$ (Figure 1.2 (b)) it approaches two in an oscillatory manner; and

- when $r = 2.5$ (Figure 1.2 (c)) it oscillates around two.

The above numerical results raise the following questions:

- can we develop tools that allow us to qualitatively describe how the model solutions depend on model parameters?
- are the there certain families of solutions with qualitatively similar behaviours?
- how do such families of solutions depend on model parameters?

We will return to analysis of the Ricker model in Worksheet 1.

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

r_1=0.5
r_2=1.5
r_3=2.5
K=2.0

T=20
t = np.arange(0, T, 1)
N_1 = np.zeros_like(t,dtype=float)
N_2 = np.zeros_like(t,dtype=float)
N_3 = np.zeros_like(t,dtype=float)

def rhs(x,r,K):

  f=x*np.exp(r*(1-x/K))
  return f


def SolveSingleDiff(t,rhs,N_0,r,K):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r,K)
  return N



N_1=SolveSingleDiff(t,rhs,N_0,r_1,K)
```

```
N_2=SolveSingleDiff(t,rhs,N_0,r_2,K)
N_3=SolveSingleDiff(t,rhs,N_0,r_3,K)


fig, ax = plt.subplots(1,3)
ax[0].plot(t, N_1)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
ax[1].plot(t, N_2)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
ax[2].plot(t, N_3)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
plt.show()
```

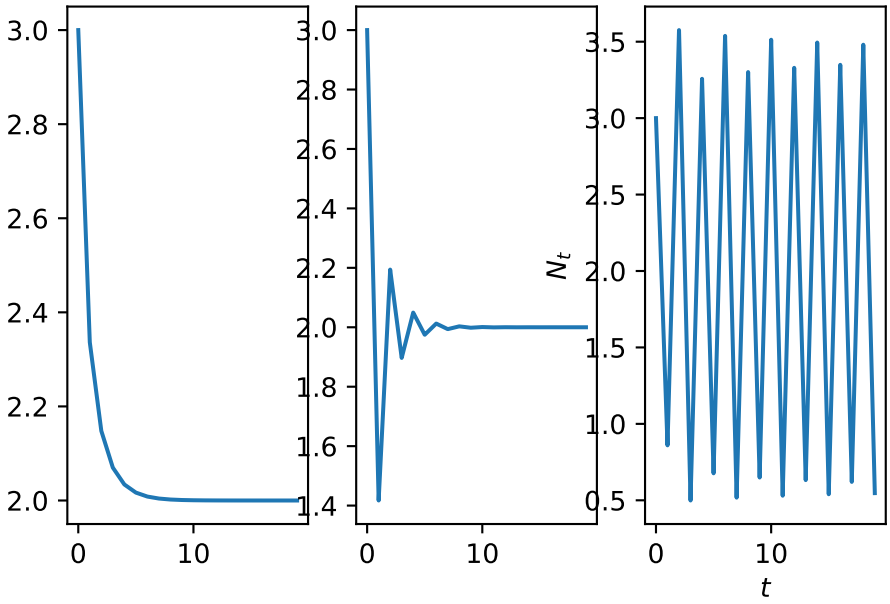

Figure 1.2: A plot of numerical solutions of the Ricker model. (a)r=0.5. (b) r=1.5. (c) r=2.5.

1.4 General techniques for analysing nonlinear difference equations

In this section we develop techniques that will be used to analyse first order, discrete-time models of the form

$$N_{t+1} = N_t f(N_t) = H(N_t). \tag{1.4}$$

1.4.1 Computational solutions

Perhaps the most obvious way to investigate an equation of the form of Equation 1.4 is to iteratively compute N_t over a prescribed time range.

Hence given some initial population N_0, the solution set $\{N_0, N_1, N_2, ..., N_T\}$ is computed up to some end time T. Such an approach provides a numerical solution for a given parameter set and initial condition. However, it does not provide much insight into the general behaviour of model.

1.4.2 Fixed points

We define N^* to be a fixed point of Equation 1.4 if and only if

$$N^* = N^* f(N^*) = H(N^*).$$

Hence the fixed points can be identified by solving an algebraic equation.

1.4.3 Linear stability

By linearising about the identified fixed points using a Taylor series expansion, it can be determined whether or not small perturbations around the fixed point grow or decay in subsequent iterations.

Making a change of dependent variables

$$N_t = N^* + \hat{N}_t,$$

Equation 1.4 transforms upon substitution to

$$N^* + \hat{N}_{t+1} = H(N^* + \hat{N}_t).$$

Taylor expanding the right-hand side yields

$$N^* + \hat{N}_{t+1} = H(N^*) + H'(N^*)\hat{N}_t + h.o.t..$$

Noting that at the fixed point

$$N^* = H(N^*),$$

cancellation yields

$$\hat{N}_{t+1} = H'(N^*)\hat{N}_t + h.o.t.$$

Hence considering only small perturbations about the fixed point, the leading order behaviour of the perturbation is governed by

$$\hat{N}_t = (H'(N^*))^t \hat{N}_0.$$

Hence the linear stability of the fixed point is governed by the sign and magnitude of the term $H'(N^*)$. If

$$|H'(N^*)| > 1,$$

the fixed point is unstable. Otherwise it it stable.

1.4.4 Cobwebbing and linear stability

A cobweb diagram allows solutions of a first order difference equation to be sketched without explicitly solving.

1.4.4.1 An algorithm for generating a cobweb diagram

The cobweb diagram is created as follows:

- Plot the t^{th} iterate, N_t, on the x axis and the $t+1^{st}$, N_{t+1} on the y axis.
- Sketch the net growth rate function $H(N_t)$.
- Sketch the line $N_{t+1} = N_t$.
- Note any intersections between the straight line and the graph of $H(N_t)$ are fixed points.
- Plot the first point $(N_0, H(N_0))$ on the cobweb diagram.
- Plot a horizontal line between $(N_0, H(N_0))$ and $(H(N_0), H(N_0))$. Note $N_1 = H(N_0)$.
- Plot a vertical line between (N_1, N_1) and $(N_1, H(N_1))$. *Repeat steps 6 and 7.

Plotting cobweb diagrams in the vicinity of a fixed point N^* for different values of $H'(N^*)$ allows us to see graphically how the derivative of the right-hand side affects linear stability of a fixed point (see Figure **??**). In particular,

- when $H'(N^*) > 1$, the fixed point is monotonically unstable;
- when $0 < H'(N^*) < 1$, the fixed point is monotonically stable;
- when $-1 < H'(N^*) < 0$, the fixed point is oscillatory stable;
- when $H'(N^*) = -1$, the fixed point is oscillatory; and
- when $H'(N^*) < -1$, the fixed point is oscillatory unstable.

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

T=10

t = np.arange(0, T, 1)
N = np.zeros_like(t,dtype=float)
N_2 = np.zeros_like(t,dtype=float)

N_0=2.4
r=0.5
r_2=1.5
N_max=2.5
N_min=1.5
def rhs(x,r):
  f=r*x+1
  return f

def SolveSingleDiff(t,rhs,N_0,r):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0


  num_time_steps=t.shape[0]

  CobwebSol=np.zeros((2*num_time_steps,2))
  CobwebSol[0,0]=N_0
  CobwebSol[0,1]=rhs(N_0,r)
  CobwebSol[1,0]=CobwebSol[0,1]
  CobwebSol[1,1]=CobwebSol[0,1]

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r)

      sol_temp=N[i-1]
      rhs_temp=rhs(sol_temp,r)
      CobwebSol[2*i,0]=sol_temp
      CobwebSol[2*i,1]=rhs_temp
      CobwebSol[2*i+1,0]=rhs_temp
      CobwebSol[2*i+1,1]=rhs_temp
```

```
  return N, CobwebSol

N,CobwebSol=SolveSingleDiff(t,rhs,N_0,r)

N_plot=np.linspace(N_min,N_max,100)
H_N_plot=rhs(N_plot,r)

fig, ax = plt.subplots(3,2)
ax[0,0].plot([N_min, N_max], [N_min, N_max])
ax[0,1].plot([N_min, N_max], [N_min, N_max])
ax[0,1].plot(N_plot, H_N_plot)
ax[1,0].plot([N_min, N_max], [N_min, N_max])
ax[1,0].plot(N_plot, H_N_plot)
ax[1,0].plot(CobwebSol[0,0], CobwebSol[0,1],'*')
ax[1,1].plot([N_min, N_max], [N_min, N_max])
ax[1,1].plot(N_plot, H_N_plot)
ax[1,1].plot(CobwebSol[0:5,0], CobwebSol[0:5,1])
ax[2,0].plot([N_min, N_max], [N_min, N_max])
ax[2,0].plot(N_plot, H_N_plot)
ax[2,0].plot(CobwebSol[0:7,0], CobwebSol[0:7,1])
ax[2,1].plot([N_min, N_max], [N_min, N_max])
ax[2,1].plot(N_plot, H_N_plot)
ax[2,1].plot(CobwebSol[:,0], CobwebSol[:,1])

plt.xlabel('$N_t$')
plt.ylabel('$N_{t+1}$')
plt.show()
```

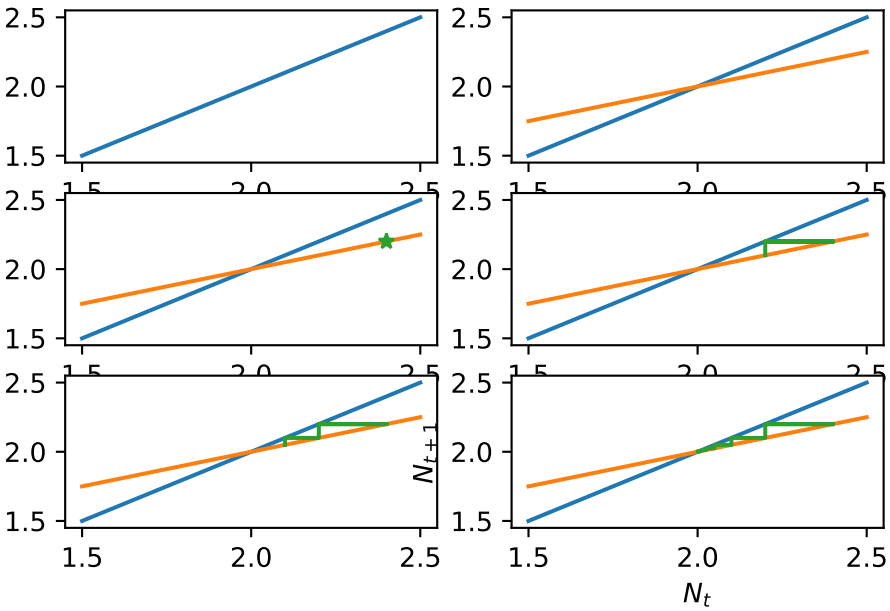

Figure 1.3: Generating a cobweb plot.

1.4.5 Bifurcation diagrams

A bifurcation is a qualitative change in solution behaviour (either a change in the number of fixed points or their stability). A bifurcation diagram is a compact means of describing bifurcations as a function of model parameters. Usually, the fixed point value of the population (i.e. N^*) is plotted against a model parameter of interest. From this diagram one can immediately see how the number of fixed points and their stability changes as a given parameter increases.

1.4.6 Perform a qualitative analysis on the Matlhusian model

Let's work through the various concepts introduced above using the Malthusian model (Equation 1.2).

The fixed points satisfy

$$N^* = rN^*.$$

As $r > 0$ the only solution is $N^* = 0$.

In this case

$$H'(N_t) = r.$$

Hence the linear stability of the solution depends on the value of the parameter r For $r > 1$ the fixed point $N^* = 0$ is monotonically unstable. For $r < 1$ the fixed point $N^* = 0$ is monotonically unstable. As expected this result is consistent with the simulation results in Figure 1.1.

For cobweb diagram:

- Sketch axes and label with N_t and N_{t+1}.
- From linear stablity analysis note there are two qualitatively distinct cases ($r < 1$ and $r > 1$). We will need a cobweb diagram for each case.
- Case 1: graph the function $H(N_t)$. In this case it is the straight line $N_{t+1} = rN_t$ with $r < 1$.
- Case 2: graph the function $H(N_t)$. In this case it is the straight line $N_{t+1} = rN_t$ with $r > 1$.
- Fill in the cobwebbed trajectories with some arbitraily chosen initial condition.
- Establish that behaviour of the cobweb is consistent with linear stability analysis.

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

T=10

t = np.arange(0, T, 1)
N = np.zeros_like(t,dtype=float)
N_2 = np.zeros_like(t,dtype=float)

N_0=3.0
r=0.5
r_2=1.5
N_max=4.0

def rhs(x,r):
  f=r*x
  return f

def SolveSingleDiff(t,rhs,N_0,r):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0


  num_time_steps=t.shape[0]

  CobwebSol=np.zeros((2*num_time_steps,2))
```

```
  CobwebSol[0,0]=N_0
  CobwebSol[0,1]=rhs(N_0,r)
  CobwebSol[1,0]=CobwebSol[0,1]
  CobwebSol[1,1]=CobwebSol[0,1]

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r)

      sol_temp=N[i-1]
      rhs_temp=rhs(sol_temp,r)
      CobwebSol[2*i,0]=sol_temp
      CobwebSol[2*i,1]=rhs_temp
      CobwebSol[2*i+1,0]=rhs_temp
      CobwebSol[2*i+1,1]=rhs_temp
  return N, CobwebSol

N,CobwebSol=SolveSingleDiff(t,rhs,N_0,r)

N_plot=np.linspace(0,N_max,100)
H_N_plot=rhs(N_plot,r)

fig, ax = plt.subplots(1,2)
ax[0].plot(t, N)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
ax[1].plot(CobwebSol[:,0], CobwebSol[:,1])
ax[1].plot(CobwebSol[0,0], CobwebSol[0,1],'*')
ax[1].plot([0, N_max], [0, N_max])
ax[1].plot(N_plot, H_N_plot)


plt.xlabel('$N_t$')
plt.ylabel('$N_{t+1}$')
plt.show()
```

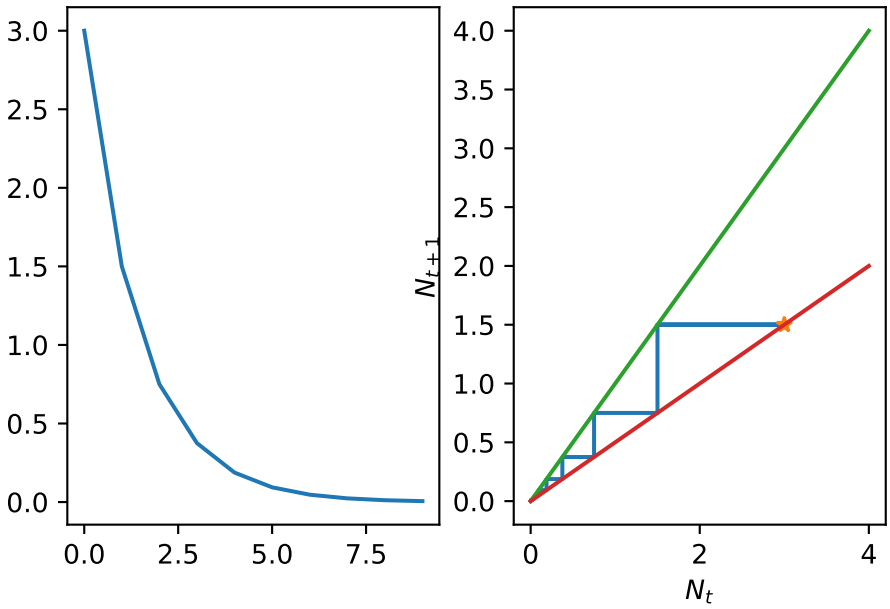

Figure 1.4: A cobweb plot of the Malthusian model solution when r=0.5. (a) Time series and (b) Cobweb diagrams.

For bifurcation diagram:

- We will plot the value of fixed point against a parameter of interest. In this case N^* against r.
- Deduce from fixed point and linear stability analysis whether there are any changes in solution behaviour as r varies (e.g. number of solution or their linear stability).
- In this case the value of the fixed point is independent of r but the the linear stability changes at $r = 1$. This is a bifurcation.
- Use annotation to represent qualitative solution behaviour.

1.5 A nonlinear model of population dynamics in discrete time

Consider a model of population growth given by

$$N_{t+1} = \frac{\gamma N_t}{1 + N_t^2}, \tag{1.5}$$

where $\gamma \in \mathfrak{R}^+$ is the linear growth rate.

1.5.1 Numerical solution

In Figure 1.5 time series solution of the model are plotted for parameter values $\gamma = 0.5$, $\gamma = 1.5$, $\gamma = 2.5$. Note qualitative changes in model behaviour in the different parameter regimes. Can you identify the fixed points? Are the solutions oscillatory?

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

T=10

t = np.arange(0, T, 1)

N_0=3.0
gam_1=0.5
gam_2=1.5
gam_3=2.5

def rhs(x,r):
  f=r*x/(1+x**2)
  return f

def SolveSingleDiff(t,rhs,N_0,r):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r)
  return N

N_1=SolveSingleDiff(t,rhs,N_0,gam_1)
N_2=SolveSingleDiff(t,rhs,N_0,gam_2)
N_3=SolveSingleDiff(t,rhs,N_0,gam_3)


fig, ax = plt.subplots(1,3)
ax[0].plot(t, N_1)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
ax[1].plot(t, N_2)
plt.xlabel('$t$')
```

```
plt.ylabel('$N_t$')
ax[2].plot(t, N_3)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
plt.show()
```

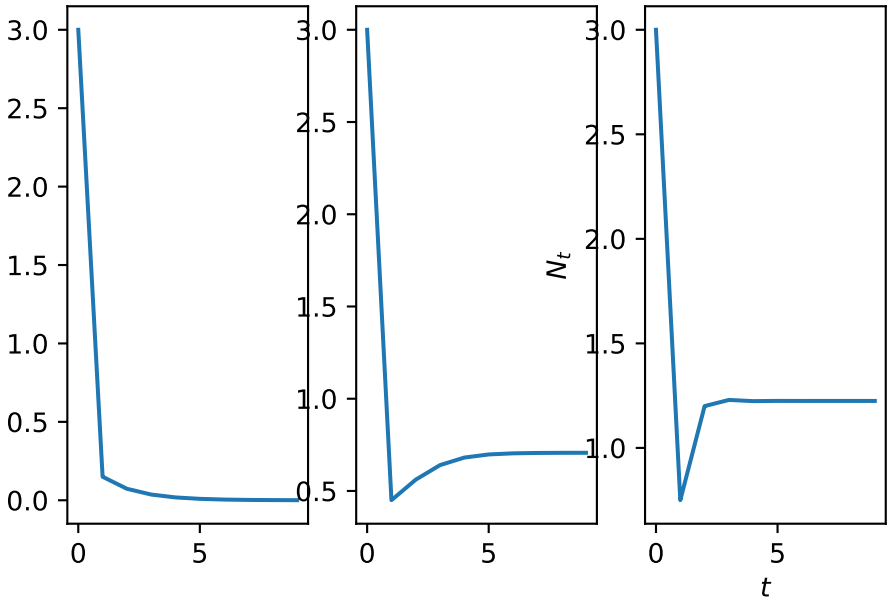

Figure 1.5: Time series solution for (a) g=0.5 and (b) g=1.5 and (c) g=10.5.

1.5.2 Fixed points

The fixed points satisfy the algebraic equation

$$N^* = \frac{\gamma N^*}{1 + {N^*}^2}$$

which has solutions

$$N^* = 0 \quad \text{and} \quad N^* = \sqrt{\gamma - 1}.$$

Sketch a graph of $H(N_t)$ against N_t in the cases $\gamma < 1$ and $\gamma > 1$.

Note that the per capita growth rate is described by the function

$$f(N_t) = \frac{\gamma}{1 + N_t^2}.$$

Hence at large populations the per capita growth rate tends to zero whilst for small populations it tends to γ. Hence so long as $\gamma > 1$ we will have net growth for small populations and net removal for large populations. The net growth rate is given by

$$H(N_t) = \frac{\gamma N_t}{1 + N_t^2}.$$

1.5.3 Parameter restrictions for biological validity

If $\gamma < 1$, there is only one biologically relevant fixed point. If $\gamma > 1$ there are two fixed-points, $N^* = 0$ and $N^* = \sqrt{\gamma - 1}$. Hence the value of the model parameter γ qualitatively affects the behaviour and number of solutions.

Sketch a graph of N^* against γ.

Roots: 0 TP:

1.5.4 Linear stability analysis

For the given model we compute

$$H'(N_t) = \frac{\gamma}{1 + N_t^2} + \frac{-2\gamma N_t^2}{(1 + N_t^2)^2}.$$

1.5.4.1 $N^* = 0$

The stability of the fixed point $N^* = 0$ is determined by

$$H'(0) = \gamma.$$

Hence if $ <1$ the first fixed point is monotonically stable as $0 < H'(0) < 1$.

When $\gamma > 1$ the second fixed point becomes biologically relevant and the first fixed point becomes monotonically unstable.

1.5.4.2 $N^* = \sqrt{\gamma - 1}$

The stability of the second fixed point, $N^* = \sqrt{\gamma - 1}$, is determined by

Hence if N^* were monotonically unstable,

$$H'(\sqrt{\gamma - 1}) = \frac{2}{\gamma} - 1 > 1,$$

and

$$\gamma < 1.$$

As a necessary condition for the biological relevance of N^* is that $\gamma > 1$ we obtain a contradiction. Hence N^*, if it is biologically relevant, cannot be monotonically unstable.

Suppose N^* is monotonically stable. Then

$$0 < H'(N^*) < 1$$

which upon substitution yields

$$0 < \frac{2}{\gamma} - 1 < 1.$$

Hence $1 < \gamma < 2$. Suppose N^* is oscillatory stable. Then

$$-1 < H'(N^*) < 0$$

which upon substitution yields

$$-1 < \frac{2}{\gamma} - 1 < 0.$$

Hence $\gamma > 2$. Hence the linear stability of the fixed point $N^* = \sqrt{\gamma - 1}$ depends on whether γ lies in the range $[0, 1]$, $[1, 2]$ or $[2, \infty]$.

Let's annotate the diagram in the previous figure with the info. obtained from the linear stability analysis. Hence the interval $[0, \infty]$ can be divided into distinct subintervals within which the linear stability of N^* is conserved. The boundaries of subintervals are bifurcation values where a the stability of the fixed point changes.

1.5.5 Cobweb diagrams

In Figure 1.6 cobweb diagrams illustrate model behaviour in the three different parameter regimes. Note that the cobweb diagrams can be sketched by hand, given an accurate enough sketch of the right-hand side function $H(N_t)$. Note also that the cobweb diagrams are consistent with the linear stability analysis but that the linear approximation is only valid close to the equilibrium point.

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

T=10

t = np.arange(0, T, 1)
N = np.zeros_like(t,dtype=float)
N_2 = np.zeros_like(t,dtype=float)
```

```
N_0=0.2
r_1=0.5
r_2=1.5
r_3=2.5
r_4=3.5

N_max=4.0

def rhs(x,r):
  f=r*x/(1+x**2)
  return f

def SolveSingleDiff(t,rhs,N_0,r):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0


  num_time_steps=t.shape[0]

  CobwebSol=np.zeros((2*num_time_steps,2))
  CobwebSol[0,0]=N_0
  CobwebSol[0,1]=rhs(N_0,r)
  CobwebSol[1,0]=CobwebSol[0,1]
  CobwebSol[1,1]=CobwebSol[0,1]

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r)

      sol_temp=N[i-1]
      rhs_temp=rhs(sol_temp,r)
      CobwebSol[2*i,0]=sol_temp
      CobwebSol[2*i,1]=rhs_temp
      CobwebSol[2*i+1,0]=rhs_temp
      CobwebSol[2*i+1,1]=rhs_temp
  return N, CobwebSol

N1,CobwebSol1=SolveSingleDiff(t,rhs,N_0,r_1)
N2,CobwebSol2=SolveSingleDiff(t,rhs,N_0,r_2)
N3,CobwebSol3=SolveSingleDiff(t,rhs,N_0,r_3)
N4,CobwebSol4=SolveSingleDiff(t,rhs,N_0,r_4)
```

```
N_plot=np.linspace(0,N_max,100)
H_N_plot1=rhs(N_plot,r_1)
H_N_plot2=rhs(N_plot,r_2)
H_N_plot3=rhs(N_plot,r_3)
H_N_plot4=rhs(N_plot,r_4)

fig, ax = plt.subplots(2,2)

ax[0,0].plot(CobwebSol1[:,0], CobwebSol1[:,1])
ax[0,0].plot(CobwebSol1[0,0], CobwebSol1[0,1],'*')
ax[0,0].plot([0, N_max], [0, N_max])
ax[0,0].plot(N_plot, H_N_plot1)

ax[0,1].plot(CobwebSol2[:,0], CobwebSol2[:,1])
ax[0,1].plot(CobwebSol2[0,0], CobwebSol2[0,1],'*')
ax[0,1].plot([0, N_max], [0, N_max])
ax[0,1].plot(N_plot, H_N_plot2)

ax[1,0].plot(CobwebSol3[:,0], CobwebSol3[:,1])
ax[1,0].plot(CobwebSol3[0,0], CobwebSol3[0,1],'*')
ax[1,0].plot([0, N_max], [0, N_max])
ax[1,0].plot(N_plot, H_N_plot3)

ax[1,1].plot(CobwebSol4[:,0], CobwebSol4[:,1])
ax[1,1].plot(CobwebSol4[0,0], CobwebSol4[0,1],'*')
ax[1,1].plot([0, N_max], [0, N_max])
ax[1,1].plot(N_plot, H_N_plot4)


plt.xlabel('$N_t$')
plt.ylabel('$N_{t+1}$')
plt.show()
```

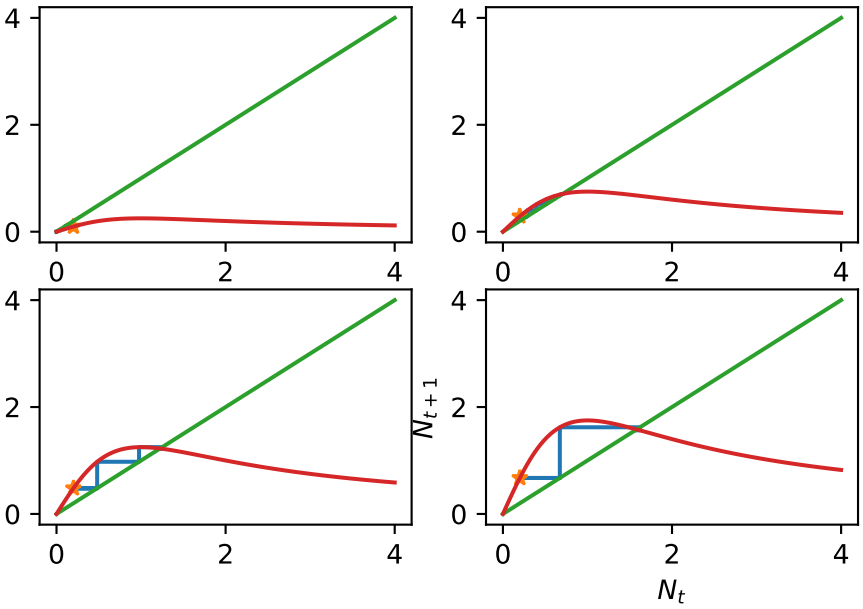

Figure 1.6: Cobweb diagrams for different values of gamma.

1.5.6 Bifurcation diagram

Bifurcations at the critical values $\gamma = 1$ and $\gamma = 2$ are highlighted in the bifurcation diagram presented in Figure 1.7. Note that the bifurcation diagram allows classification of the different model behaviours in a single plot without explicitly calculating the solution to the model.

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



gam_plot=np.linspace(0,4,100)
N_star_1=0*np.zeros_like(gam_plot)

gam_plot2=np.linspace(1,4,100)
N_star_2=np.sqrt(gam_plot2-1)

fig, ax = plt.subplots(1)

ax.plot(gam_plot,N_star_1)
ax.plot(gam_plot2,N_star_2)
```

```
plt.xlabel('$gamma$')
plt.ylabel('$N^*$')
plt.show()
```

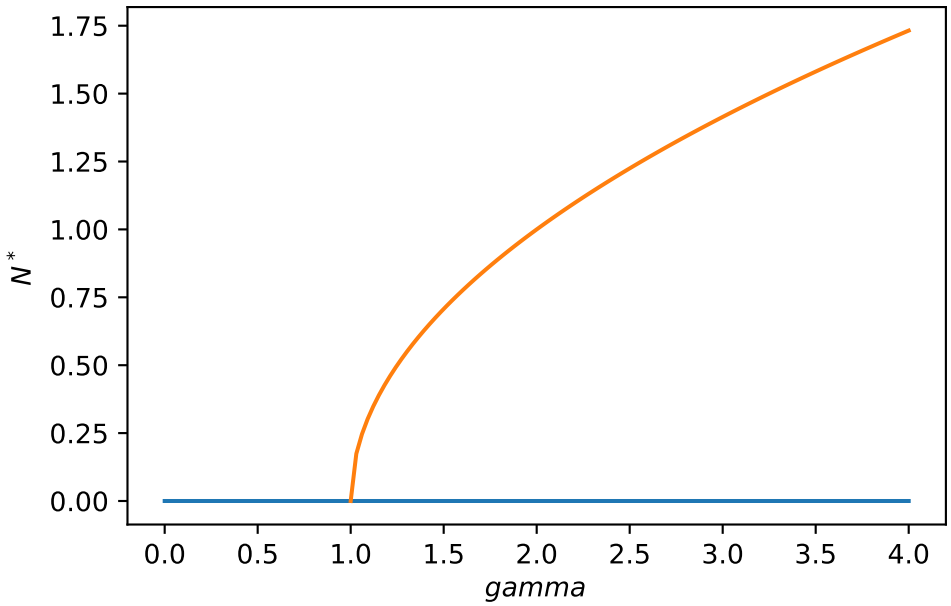

Figure 1.7: A bifiurcation diagram.

1.6 An application - how much harvesting can a population sustain?

Models of population dynamics can be used to study how interventions will affect population dynamics. Extending from the model developed in the previous example, a valid question might be what is the maximal rate of harvesting a fish stock can sustain without becoming extinct? ### Including a harvesting term Introducing a harvesting term at per capita harvesting rate h, the governing model equation becomes

$$N_{t+1} = \frac{\gamma N_t}{1 + N_t^2} - hN_t, \tag{1.6}$$

and the questions we want to ask are: (i) does the introduction of harvesting change the dynamics of the system?; and (ii) what rate of harvesting such a population could withstand?

1.6.1 Direct simulation

Based on the previous analysis we consider the system in a parameter regime where without harvesting there is a stable fixed point ($\gamma > 1$).

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

T=10

t = np.arange(0, T, 1)

N_0=0.1
gam=5.0
h_1=0.1
h_2=1.0
h_3=3.0

def rhs(x,r,h):
  f=r*x/(1+x**2)-h*x
  return f

def SolveSingleDiff(t,rhs,N_0,r,h):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r,h)
  return N

N_1=SolveSingleDiff(t,rhs,N_0,gam,h_1)
N_2=SolveSingleDiff(t,rhs,N_0,gam,h_2)
N_3=SolveSingleDiff(t,rhs,N_0,gam,h_3)


fig, ax = plt.subplots(1,3)
ax[0].plot(t, N_1)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
ax[1].plot(t, N_2)
plt.xlabel('$t$')
```

```
plt.ylabel('$N_t$')
ax[2].plot(t, N_3)
plt.xlabel('$t$')
plt.ylabel('$N_t$')
plt.show()
```

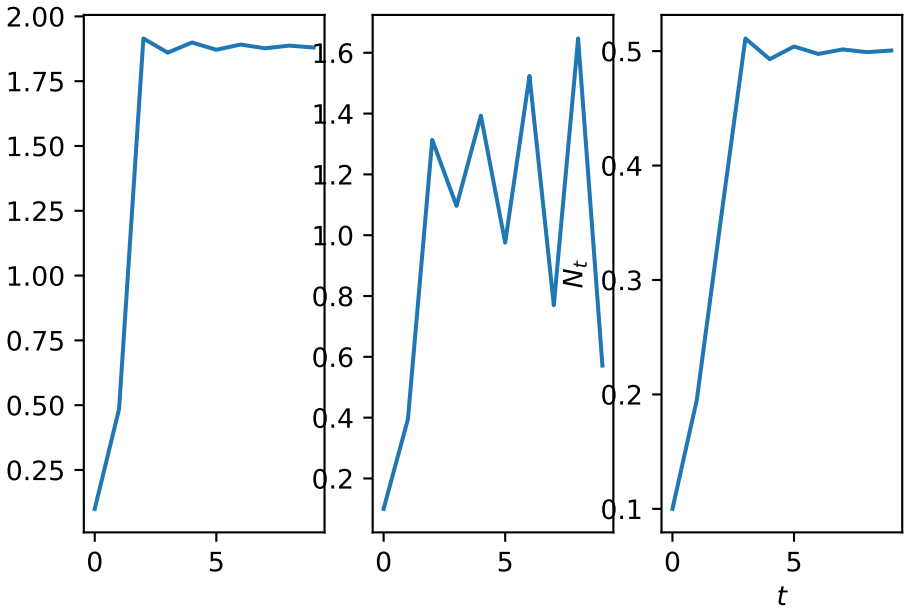

Figure 1.8: Time series solution for (a) g=0.5 and (b) g=1.5 and (c) g=10.5.

In Figure 1.8 time series solutions at increasing harvesting rates ($h = 0.1$, $h = 1$, $h = 3$) are presented. For low harvesting rates the system behaves almost identically to the no harvesting case but with an expected reduction in the fixed point value. However for intermediate harvesting the population undergoes oscillations. However, for further increased harvesting rates there appears again to be a stable fixed point.

Can analysis of the model help us understand how/why changes in solutions occur?

1.6.2 Fixed points

The fixed points of the system satisfy

$$N^* = \frac{\gamma N^*}{1 + {N^*}^2} - hN^*.$$

Hence the fixed points are

$$N^* = 0$$

and

$$N^* = \sqrt{\frac{\gamma}{1+h} - 1}.$$

Note that harvesting lowers the population size of the non-zero fixed point (when it exists).

Can we deduce a condition that must hold on the parameters γ and h in order that there is a non trivial biologically relevant fixed point?

Requiring

$$N^* > 0$$

implies

$$\frac{\gamma}{1+h} - 1 > 0$$

.

1.6.3 Linear stability

The linear stability is determined by

$$H'(N_t) = \frac{\gamma}{1+{N_t}^2} - \frac{2\gamma N_t^2}{(1+{N_t}^2)^2} - h.$$

At $N^* = 0$ we obtain

$$H'(0) = \gamma - h.$$

Hence if $\gamma > 1 + h$, $N^* = 0$ is unstable.

Note that this is the condition that determines whether the non-zero fixed point exists or not.

At

$$N^* = \sqrt{\frac{\gamma}{1+h} - 1},$$

$$H'\left(\sqrt{\frac{\gamma}{1+h} - 1}\right) = \frac{\gamma}{1+\frac{\gamma}{1+h} - 1} - \frac{2\gamma\frac{\gamma}{1+h} - 1}{(1+\frac{\gamma}{1+h} - 1)^2} - h.$$

Hence linear stability is a function of two parameters γ and h.

We can show that

$$H'(N^*) = h^2\frac{2}{\gamma} + h\frac{2}{\gamma}(2-\gamma) + \frac{2}{\gamma} - 1,$$

{#harvestingH_prime} and hence that when $h = 0$ we retrieve the stability condition from the previous model.

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

h_vec=np.linspace(0.1,5,100)
gamma_1=(1+h_vec)**2/h_vec
gamma_2=1+h_vec
gamma_3=h_vec

fig, ax = plt.subplots(1)
ax.plot(h_vec, gamma_1)
ax.plot(h_vec, gamma_2)
ax.plot(h_vec, gamma_3)

plt.xlabel('$h$')
plt.ylabel('$\gamma$')
plt.show()
```

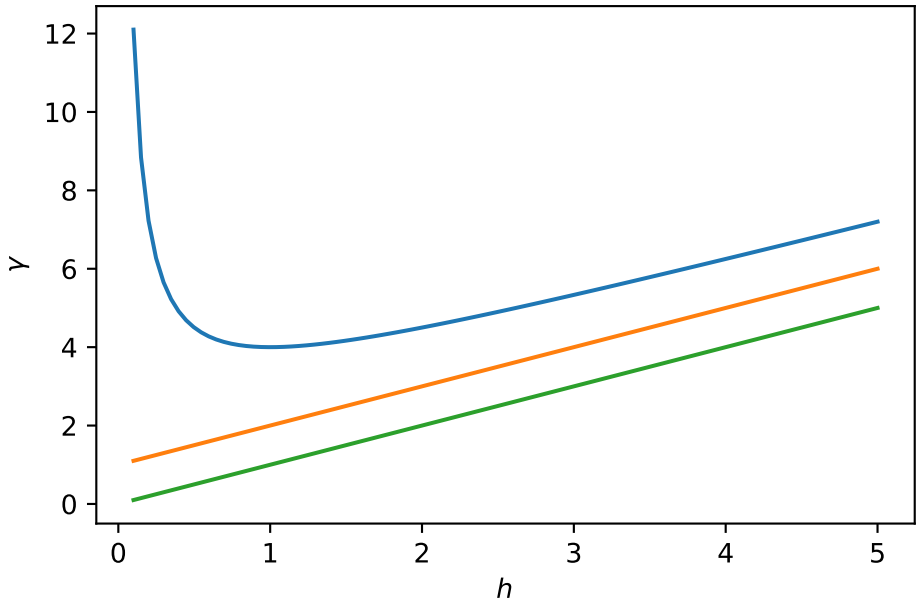

Figure 1.9: Stability regions for the harvesting model.

1.6.4 Stability boundaries in the $h\gamma$ plane

The contour at $H'(N^*) = 1$ can be identified by solving

$$h^2\frac{2}{\gamma} + h\frac{2}{\gamma}(2-\gamma) + \frac{2}{\gamma} - 1 = 1,$$

yielding

$$\gamma = 1 + h.$$

The contour at $H'(N^*) = -1$ can be represented by $

$$h^2\frac{2}{\gamma} + h\frac{2}{\gamma}(2-\gamma) + \frac{2}{\gamma} - 1 = -1,$$

Solving for γ yields

$$\gamma = \frac{(1+h)^2}{h}.$$

In Figure Figure 1.9 we plot the contours of the hypersurface $H'(N^*)$ at the critical values of -1, 0 and 1. Note that the points in parameter space used to generate the simulation results in Figure **??** are $(h, \gamma) = (0.1, 5)$ $(h, \gamma) = (1, 5)$ and $(h, \gamma) = (3, 5)$. Cobweb diagrams in different regions of parameter space are presented in Figure **??**. The plot in Figure 1.9 can be used to explain why the model transfers from a stable fixed point, through oscillatory fixed point and back to a stable fixed point as the harvesting rate increases from 0.1, to 1 and then 5?

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

T=10

t = np.arange(0, T, 1)
N = np.zeros_like(t,dtype=float)
N_2 = np.zeros_like(t,dtype=float)

N_0=0.2
h_1=0.1
h_2=1.0
h_3=3.0
r=5.0

N_max=4.0

def rhs(x,r,h):
  f=r*x/(1+x**2)-h*x
```

```
  return f

def SolveSingleDiff(t,rhs,N_0,r,h):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0

  num_time_steps=t.shape[0]

  CobwebSol=np.zeros((2*num_time_steps,2))
  CobwebSol[0,0]=N_0
  CobwebSol[0,1]=rhs(N_0,r,h)
  CobwebSol[1,0]=CobwebSol[0,1]
  CobwebSol[1,1]=CobwebSol[0,1]

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r,h)

      sol_temp=N[i-1]
      rhs_temp=rhs(sol_temp,r,h)
      CobwebSol[2*i,0]=sol_temp
      CobwebSol[2*i,1]=rhs_temp
      CobwebSol[2*i+1,0]=rhs_temp
      CobwebSol[2*i+1,1]=rhs_temp
  return N, CobwebSol

N1,CobwebSol1=SolveSingleDiff(t,rhs,N_0,r,h_1)
N2,CobwebSol2=SolveSingleDiff(t,rhs,N_0,r,h_2)
N3,CobwebSol3=SolveSingleDiff(t,rhs,N_0,r,h_3)

N_plot=np.linspace(0,N_max,100)
H_N_plot1=rhs(N_plot,r,h_1)
H_N_plot2=rhs(N_plot,r,h_2)
H_N_plot3=rhs(N_plot,r,h_3)

fig, ax = plt.subplots(1,3)

ax[0].plot(CobwebSol1[:,0], CobwebSol1[:,1])
ax[0].plot(CobwebSol1[0,0], CobwebSol1[0,1],'*')
ax[0].plot([0, N_max], [0, N_max])
ax[0].plot(N_plot, H_N_plot1)
```

```
ax[1].plot(CobwebSol2[:,0], CobwebSol2[:,1])
ax[1].plot(CobwebSol2[0,0], CobwebSol2[0,1],'*')
ax[1].plot([0, N_max], [0, N_max])
ax[1].plot(N_plot, H_N_plot2)

ax[2].plot(CobwebSol3[:,0], CobwebSol3[:,1])
ax[2].plot(CobwebSol3[0,0], CobwebSol3[0,1],'*')
ax[2].plot([0, N_max], [0, N_max])
ax[2].plot(N_plot, H_N_plot3)



plt.xlabel('$N_t$')
plt.ylabel('$N_{t+1}$')
plt.show()
```

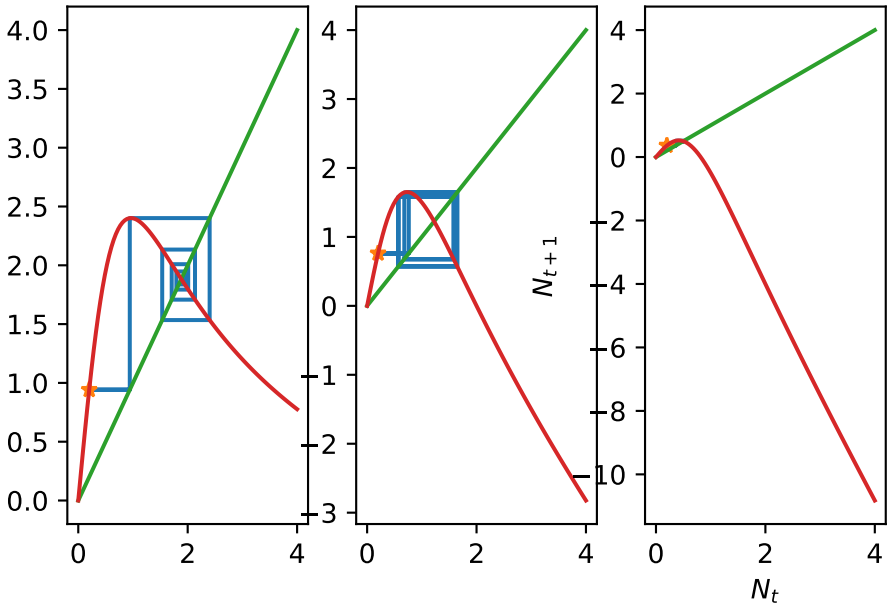

Figure 1.10: Cobweb diagrams for different values of h.

We can construct a cobweb diagram for the case $(h, \gamma) = (3, 5)$.

1.7 A note on the modelling of real-world fish stocks

You can find reports on fish stocks (measurements and modelling work) from the International Council for the Exploration of the Sea at the link http://www.ices.dk/Pages/default.aspx. Here's a link to the data for Atlantic salmon https://tinyurl.com/ya8sh72j. Note that although the models we have worked on are not detailed enough to accurately model real fish population dynamics, many of the principles we have covered arise in the cutting edge models.

1.8 Oscillations

Linear stability analysis describes the evolution of small perturbations close to a fixed point. But far from a fixed point the nonlinear terms that were dropped in a Taylor expansion are no longer negligible. In particular, in the case where $H'(N^*) < -1$, it can be the case that regions of parameter space in which a fixed point is oscillatory unstable can give rise to periodic and chaotic solutions.
Defining a periodic solution Consider the general form

$$N_{t+1} = H(N_t) \tag{1.7}$$

A solution to Equation 1.7 isdefined to be periodic with period T if

$$N_{t+T} = N_t \quad \forall t, N_{t+\tau} \quad \neq N_t \quad \forall t, \quad \tau < T.$$

Period 2 solutions can be identified by looking for solutions that repeat after two iterations. Suppose $\bar{N}$ is a period 2 solution of Equation 1.7 and let $\bar{N} = N_1$. Then

$$N_2 = H(\bar{N}).$$

However,

$$N_3 = H(N_2) = H(H(\bar{N}))$$

and if $\bar{N}$ is a period 2 solution $N_3 = N_1 = \bar{N}$. Hence period 2 solutions can be calculated by solving the algebraic equation

$$\bar{N} = H(H(\bar{N})).$$

Note that period two solutions are fixed points of the problem

$$N_{t+2} = H^2(N_t) = g(N_t).$$

Using the tools we have developed, period solutions and their stability can be determined. Furthermore, longer period solutions can be identified by generalising the argument but at the expense of ever increasingly complicated right-hand side functions. In many systems, such as the harvesting model and the logistic equation, increasing the growth rate parameter leads

initially to the fixed point becoming unstable and the emergence of period two solutions, then period 4 solutions and so on until eventually there is a transition to chaotic solutions (see, for example, Figure 1.11).

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

T=40

t = np.arange(0, T, 1)
N = np.zeros_like(t,dtype=float)
N_2 = np.zeros_like(t,dtype=float)

N_0=0.1
r_1=9.68
r_2=12.221
r_3=30.25
h=0.1

N_max=20.0

def rhs(x,r,h):
  f=r*x/(1+x**2)-h*x
  return f

def SolveSingleDiff(t,rhs,N_0,r,h):
  N = np.zeros_like(t,dtype=float)
  N[0]=N_0


  num_time_steps=t.shape[0]

  CobwebSol=np.zeros((2*num_time_steps,2))
  CobwebSol[0,0]=N_0
  CobwebSol[0,1]=rhs(N_0,r,h)
  CobwebSol[1,0]=CobwebSol[0,1]
  CobwebSol[1,1]=CobwebSol[0,1]

  for i in t:
    if i>0:
      N[i]=rhs(N[i-1],r,h)
```

```
        sol_temp=N[i-1]
        rhs_temp=rhs(sol_temp,r,h)
        CobwebSol[2*i,0]=sol_temp
        CobwebSol[2*i,1]=rhs_temp
        CobwebSol[2*i+1,0]=rhs_temp
        CobwebSol[2*i+1,1]=rhs_temp
    return N, CobwebSol

N1,CobwebSol1=SolveSingleDiff(t,rhs,N_0,r_1,h)
N2,CobwebSol2=SolveSingleDiff(t,rhs,N_0,r_2,h)
N3,CobwebSol3=SolveSingleDiff(t,rhs,N_0,r_3,h)

N_plot=np.linspace(0,N_max,100)
H_N_plot1=rhs(N_plot,r_1,h)
H_N_plot2=rhs(N_plot,r_2,h)
H_N_plot3=rhs(N_plot,r_3,h)

fig, ax = plt.subplots(3,2)

ax[0,0].plot(t, N1)

ax[0,1].plot(CobwebSol1[:,0], CobwebSol1[:,1])
ax[0,1].plot(CobwebSol1[0,0], CobwebSol1[0,1],'*')
ax[0,1].plot([0, N_max], [0, N_max])
ax[0,1].plot(N_plot, H_N_plot1)

ax[1,0].plot(t, N2)

ax[1,1].plot(CobwebSol2[:,0], CobwebSol2[:,1])
ax[1,1].plot(CobwebSol2[0,0], CobwebSol2[0,1],'*')
ax[1,1].plot([0, N_max], [0, N_max])
ax[1,1].plot(N_plot, H_N_plot2)

ax[2,0].plot(t, N3)

ax[2,1].plot(CobwebSol3[:,0], CobwebSol3[:,1])
ax[2,1].plot(CobwebSol3[0,0], CobwebSol3[0,1],'*')
ax[2,1].plot([0, N_max], [0, N_max])
ax[2,1].plot(N_plot, H_N_plot3)

plt.xlabel('$N_t$')
```

```
plt.ylabel('$N_{t+1}$')
plt.show()
```

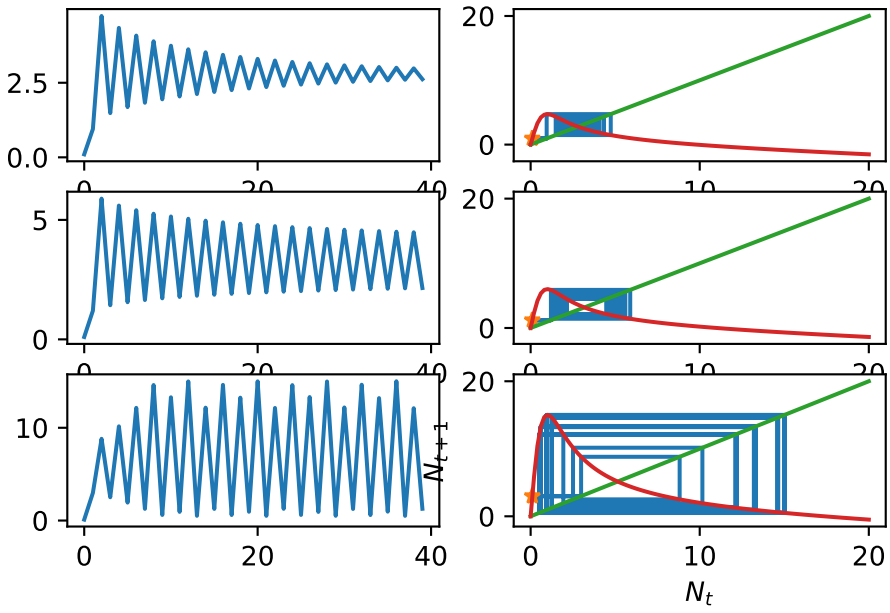

Figure 1.11: Transition form period 2 to period 4 solutions in the harvesting model

Exercise: identify an equation satisfied by period 2 solutions of the logistic map

$$N_{t+1} = rN_1(1 - N_t).$$

2 Discrete Time Two Pop

3 Multiple species in discrete time

In this section we generalise the previous approach by considering the population dynamics of two interacting populations.

3.1 A general model of two interacting species in discrete time

Let N_t and P_t represent population densities at time t where t is a discrete variable.

We consider governing equations of the form

$$N_{t+1} = f(N_t, P_t),$$
$$P_{t+1} = g(N_t, P_t).$$

where the population dynamics of the species are coupled to one another via the functions f and g.

The precise form for f and g will be defined by the biological system under study. Typical interpopulation interactions that are studied are: predator prey models, competition and cooperation.

3.2 General techniques for analysing coupled first order difference equations

3.2.1 Fixed points

The fixed points of Equation 3.2 (N^*, P^*) satisfy

$$N^* = g(N^*, P^*)$$
$$P^* = f(N^*, P^*).$$

3.2.2 Linear stability analysis

To consider linear stability of a steady state we consider the change of variable

$$N_t = N^* + \hat{N}_t,$$
$$P_t = P^* + \hat{P}_t.$$

After substitution in Equation 3.2 and making Taylor expansions about (N^*, P^*) we obtain at leading order

$$\begin{pmatrix} \hat{N}_{t+1} \\ \hat{P}_{t+1} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial N} & \frac{\partial f}{\partial P} \\ \frac{\partial g}{\partial N} & \frac{\partial g}{\partial P} \end{pmatrix}_{(N^*,P^*)} \begin{pmatrix} \hat{N}_t \\ \hat{P}_t \end{pmatrix}. \tag{3.1}$$

Note the appearance of the Jacobian matrix

$$A = \begin{pmatrix} \frac{\partial f}{\partial N} & \frac{\partial f}{\partial P} \\ \frac{\partial g}{\partial N} & \frac{\partial g}{\partial P} \end{pmatrix}_{(N^*,P^*)} \begin{pmatrix} \hat{N}_t \\ \hat{P}_t \end{pmatrix}.$$

3.2.3 Solving the linearised problem

Defining

$$\mathbf{w}_t = \begin{pmatrix} \hat{N}_t \\ \hat{P}_t \end{pmatrix},$$

$$\mathbf{w}_{t+1} = A\mathbf{w}_t$$

By solving this linear system we can investigate whether small perturbation about the fixed point grow or decay in magnitude as time evolves and hence determine the linear stability of the fixed point.

The solution of Equation 3.1 takes the form

$$\mathbf{w}_t = \sum_{i=1}^{2} C_i \lambda_i^t \mathbf{c}_i,$$

where the C_i's are constants determined by initial conditions and the λ_i's and $\mathbf{c}_i$'s are the eigenvalues and eigenvectors of A, respectively. From this form we can see that if the magnitude of all eigenvalues is less than one, i.e.

$$|\lambda_i| < 1, \qquad \forall i,$$

the fixed point is linearly stable. If at least one of the eigenvalues has $|\lambda_i| > 1$ then the fixed point is unstable to linear perturbations.

3.2.4 Jury conditions

In many cases it is not very useful to explicitly compute the eigenvalues of the Jacobian matrix. In such cases we can employ the Jury conditions in order to determine when a fixed point is linearly stable.

Recall that for a two dimensional matrix, the eigenvalues satisfy the quadratic characteristic equation

$$\lambda^2 - trA\lambda + detA = 0.$$

The stability of the fixed is guaranteed if $|\lambda_i| < 1 \ \forall i$.

Consider the characteristic equation

$$P(\lambda) = \lambda^2 + a\lambda + b = 0,$$

where $a, b \in \Re$. Note that $a = -trA$ and $b = \det A$.

The Jury conditions state that $|\lambda_i| < 1 \forall \ \ i$ if, and only if, * $b < 1$, * $1+a+b > 0$, * *$1-a+b > 0$. See Figure **??** for schematic diagram.

3.2.5 Proof of the Jury conditions

The roots of $P(\lambda)$ are

$$\lambda_{1,2} = \frac{-a \pm \sqrt{a^2 - 4b}}{2}.$$

3.2.6 Complex roots

Suppose $a^2 - 4b < 0$. The roots are complex. Since b is equal to the product of the roots, we find that

$$b = \lambda_1\lambda_2 = |\lambda_1|^2 = |\lambda_2^2|.$$

Hence $\lambda_i < 1 \ \forall \ \ i$ if and only if $b < 1$.

For the other conditions we introduce the identity:

$$a^2 - 4b = (|a| - 2)^2 - 4(1 + b - |a|).$$

Therefore, when $a^2 - 4b < 0$, the inequality requires

$$(|a| - 2)^2 - 4(1 + b - |a|) < 0.$$

This can only occur if

$$1 + b - |a| > 0 \implies 1 + b - a > 0 \ \text{ and } \ 1 + b + a > 0.$$

3.2.7 Real roots

Suppose $a^2 - 4b \geq 0$

The largest magnitude of the roots is

$$R = \max\{|\lambda_1|, |\lambda_2|\} = \frac{|a| + \sqrt{a^2 - 4b}}{2}.$$

This is an increasing function of $|a|$.

$R = 1$

$$\implies \frac{|a| + \sqrt{a^2 - 4b}}{2} = 1 \implies |a| - 2 = -\sqrt{|a|^2 - 4b} \implies |a|^2 - 4|a| + 4 = |a|^2 - 4b \implies |a| = b + 1.$$

Hence $0 \leq R < 1$ if and only if $0 \leq |a| < 1 + b$. Also, since $|\lambda_i| < 1 \forall i$ implies that $|\lambda_1 \lambda_2| < 1$, it follows that $|b| < 1$ must hold.

3.3 Host Parasitoid infection

Parasitoids are creatures that have a free living and parasitic stage. The free-living adult lays eggs in a host that later hatch and develop after eating the host. The discrete stages of the parasitoids life cycle and the dependence on its reproduction on the availability of host suggest a discrete time, multi species models. Let N_t and P_t represent the number of hosts and parasitoids at time t, respectively. Let R_0 represent the reproductive ratio of host in the absence of parasites and C the average number of viable eggs laid by each parasite on a host. ### Model equations

We consider equations of the form

$$\begin{aligned} N_{t+1} &= R_0 N_t f(N_t, P_t), \\ P_{t+1} &= C N_t (1 - f(N_t, P_t)). \end{aligned}$$

The justification for this form is that at a given time t there are N_t hosts. A total of $N_t f(N_t, P_t)$ escape the parasite and are able to reproduce whilst a total of $N_t(1 - f(N_t, P_t))$ do not escape the parasite and lead to parasitic reproduction at the next time step.

Choosing

$$f(N_t, P_t) = e^{-aP_t},$$

yields the Nicholson Bailey model

$$\begin{aligned} N_{t+1} &= R_0 N_t e^{-aP_t}, \\ P_{t+1} &= C N_t (1 - e^{-aP_t}). \end{aligned}$$

3.3.1 Fixed points

The fixed points satisfy

$$N^* = R_0 N^* e^{-aP^*},$$
$$P^* = CN^*(1 - e^{-aP^*}).$$

The first equation yields

$$N^* = 0.$$

Suppose $N^* \neq 0$

$$1 = R_0 e^{-aP^*}.$$

Hence

$$P^* = \frac{1}{a} \ln R_0.$$

Consider the second equation. Suppose $N^* = 0$. We obtain

$$P^* = 0.$$

Hence one fixed point is $(0, 0)$.

Suppose $P^* = \frac{1}{a} \ln R_0$.

$$\frac{1}{a} \ln R_0 = CN^*(1 - \frac{1}{R_0}).$$

Hence

$$N^* = \frac{\frac{1}{a} \ln R_0}{C(1 - \frac{1}{R_0})} = \frac{R_0 \ln R_0}{aC(R_0 - 1)}.$$

Hence the second fixed point is

$$\left(\frac{R_0 \ln R_0}{aC(R_0 - 1)}, \frac{1}{a} \ln R_0 \right).$$

We can verify by substitution that

$$\left(\frac{R_0 \ln R_0}{aC(R_0 - 1)}, \frac{1}{a} \ln R_0 \right).$$

is a fixed point. We can then deduce a condition on the model parameters that must hold in order that the fixed point is biologically relevant.

To verify by substitution we substitute the proposed solution into the governing equations. Note that

$$e^{-aP^*} = \frac{1}{R_0}.$$

In this case

$$\frac{R_0 \ln R_0}{aC(R_0-1)} = R_0 \frac{R_0 \ln R_0}{aC(R_0-1)} \frac{1}{R_0},$$
$$\frac{1}{a} \ln R_0 = C \frac{R_0 \ln R_0}{aC(R_0-1)} (1 - \frac{1}{R_0}).$$

Cancellation shows that both equations hold. Hence the fixed point is a valid fixed point. To be biologically relevant we require that both components of the solution are real and positive. In this case this leads to the condition $R_0 > 1\}$.

3.3.2 Linear stability

The Jacobian matrix is given by

$$A_{(N_t,P_t)} = \begin{pmatrix} R_0 e^{-aP_t} & -R_0 a N_t e^{-aP_t} \\ c(1-e^{-aP_t}) & aCN_t e^{-aP_t} \end{pmatrix}.$$

3.3.3 Linear stability of the trivial fixed point

Evaluating at (0,0) yields

$$A = \begin{pmatrix} R_0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Hence the eigenvalues are R_0 and 0. If $0 < R_0 < 1$ (0,0) is stable whilst if $R_0 > 1$ (0,0) is unstable.

3.3.4 Linear stability of the nontrivial fixed point

We can show that the Jacobian matrix evaluated at the nontrivial fixed point can be written as

$$A = \begin{pmatrix} 1 & -\frac{R_0 \ln R_0}{c(R_0-1)} \\ c(1-\frac{1}{R_0}) & \frac{\ln R_0}{R_0-1} \end{pmatrix}.$$

and deduce that the eigenvalues of A satisfy the characteristic polynomial

$$\lambda^2 - \lambda \left(1 + \frac{\ln R_0}{R_0 - 1}\right) + \frac{R_0 \ln(R_0)}{R_0 - 1} = 0.$$

3.3.5 Employing the Jury conditions

To proceed with linear stability analysis we employ the Jury conditions. Consider the polynomial

$$\lambda^2 + a\lambda + b = 0,$$

In our case

$$a = -(1 + \frac{\ln R_0}{R_0 - 1})$$

and

$$b = \frac{R_0 \ln(R_0)}{R_0 - 1}.$$

The third Jury condition ($b < 1$) implies that for linear stability

$$\ln R_0 < 1 - \frac{1}{R_0}.$$

To demonstrate that this inequality is not true for $R_0 > 1$, let $f_1 = \ln R_0$ and $f_2 = 1 - 1/R_0$. When $R_0 = 1$, $f_1 = f_2 = 0$. However,

$$f_1' = \frac{1}{R_0} \quad \text{and} \quad f_2' = \frac{1}{R_0^2},$$

implies

$$f_1' > f_2' \quad \forall \; R_0 > 1.$$

Hence

$$\ln R_0 > 1 - \frac{1}{R_0}.$$

Hence the fixed point is unstable if $R_0 > 1$. %Note that the general conditions for stability are known as the Jury conditions.

Part II

Continuous time

4 Continuous time, single species population dynamics

In this chapter we define time, t, to be a continuous variable and the population density, $N(t)$, to represent the size of a given population.

4.1 The Malthusian (linear) model

4.1.1 Deriviation

As time is now a continuous variable, we consider an interval of time δt. Consider a population of size $N(t)$. If the per capita production rate is b, then in time δt the increase in population size will be

$$bN(t)\delta t.$$

Similarly, if the per capita death rate is d, the decrease in population size in time δt will be

$$dN(t)\delta t.$$

Hence

$$N(t+\delta t) = N(t) + (bN(t) - dN(t))\delta t.$$

Rearranging and taking the limit as $\delta t \to 0$ yields

$$\frac{dN}{dt} = (b-d)N = rN(t).$$

4.1.2 Solution

The solution is given by

$$N(t) = N_0 \exp(rt).$$

We can describe qualitatively different solutions behaviours.

If $r > 0$ the solution increases exponentially with time. If $r < 0$ it decreases exponentially. When $r = 0$ the solution is constant.

Malthusian growth in the continuous time model leads to either a constant population, exponentially increasing or exponentially decreasing growth. As was the case for the discrete time model, such a model could account for biological system only in particularly limited circumstances.

The Malthusian model (constant per capita growth rate) can exhibit a limited range of behaviour. To account for limitation of population growth at large population densities due to, for example, limited resources, we consider nonlinear per capita growth rates.

4.2 Tools for analysing the dynamics of a single population in continuous time

Let $N = N(t)$. We consider the general form for a single species model defined in continuous time

$$\frac{dN}{dt} = f(N)N = H(N), \tag{4.1}$$

where f is the per capita growth rate and H is the net growth rate.

4.2.1 Numerical solution

Although an equation of the form Equation 4.1 may not be explicitly integrable, so long as the right-hand side is sufficiently well behaved, we can numerically integrate the problem (e.g. using packages such as Python 's odeint). Such a technique provides a numerical approximation to the exact solution, given specific values of model parameters and initial conditions (i.e. we can graph solutions).

4.2.2 Nondimensionalisation

We nondimensionalise a model by changing from variables that have dimensions (both dependent and independent) to variables that are dimensionless.

4.2.2.1 Dimensional analysis

The dimensions/units of each term in a model must be consistent. This observation helps to determine the units of different parameters in a model.

For example, we can deduce that the units of the parameter r in the Malthusian model

$$\frac{dN}{dt} = rN.$$

The units of the left-hand term are

$$\frac{\#popdensity}{\#time}.$$

The units of the term on the right-hand side are

$$\#r\#popdensity.$$

For dimensional consistency

$$\#r = \frac{1}{\#time}.$$

4.2.2.2 Dimensionless variables

We could nondimensionalise Equation 4.1 by defining the dimensionless variables

$$n = \frac{N}{\tilde{N}}$$
$$\tau = \frac{t}{\tilde{T}}.$$

Making the proposed change of variables in Equation 4.1 we obtain the dimensionless model

$$\frac{dn}{d\tau} = \frac{\tilde{T}}{\tilde{N}} H(\tilde{N}n(\tau)).$$

The choice for scalings is in general not unique (i.e. N^* and T^*). Appropriate choices can result in fewer dimensionless parameters in the dimensionless model.

Exercise: Show that the Mathusian model

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

can be written in nondimensional form

$$\frac{dn}{d\tau} = n.$$

Introducing dimensionless variable

$$n = \frac{N}{\tilde{N}} \quad \tau = \frac{t}{\tilde{T}}$$

yields

$$\frac{dn}{d\tau} = r\tilde{T}n$$

Choosing

$$\tilde{T} = \frac{1}{r}$$

yields

$$\frac{dn}{d\tau} = n.$$

4.2.3 Steady-state

We denote steady-state solutions of Equation 4.1 using $N = N^*$. At a steady-state,

$$\frac{dN}{dt} = 0$$

and hence steady-states can be obtained by solving the algebraic equation

$$H(N^*) = 0.$$

4.2.4 Linear stability analysis

4.2.4.1 A change of dependent variable

To perform a linear stability analysis we make the change of variables

$$N(t) = N^* + \hat{N}(t)$$

where the new dependent variable, $\hat{N}(t)$, is a perturbation about the steady state. The time derivative on the left-hand side of Equation 4.1 transforms to

$$\frac{dN}{dt} = \frac{d}{dt}(N^*) + \frac{d}{dt}(\hat{N}(t)) = \frac{d\hat{N}(t)}{dt}.$$

Hence Equation 4.1 transforms to

$$\frac{d\hat{N}(t)}{dt} = H(N^* + \hat{N}(t)).$$

4.2.4.2 Taylor expansion and a linear system

Employing the Taylor expansion on the right-hand side of Equation 4.1 and making the assumption that perturbations are small

$$\frac{d\hat{N}(t)}{dt} = H(N^*) + H'(N^*)\hat{N}(t) + H''(N^*)\hat{N}^2(t) + h.o.t.$$

Noting that

$$H(N^*) = 0$$

and retaining linear terms yields

$$\frac{d\hat{N}(t)}{dt} = H'(N^*)\hat{N}(t)$$

with solution

$$\hat{N}(t) = \eta e^{H'(N^*)t}$$

where η is some initial perturbation about the steady-state.

4.2.4.3 A condition for linear stability

When $H'(N^*) > 0$ the perturbation grows exponentially fast and the steady-state is unstable. \

When $H'(N^*) < 0$ the perturbation decays exponentially fast and the steady-state is stable.

4.2.5 Graphical solution

We can graphically identify steady-states and their stability by plotting dN/dt against $N(t)$. Steady-states arise at the roots of $H(N)$. The sign of the derivative at a root determines its linear stability.

4.2.6 Bifurcation diagrams

Bifurcations arise when the number of solutions or their stability changes at a given value of a model parameter. By plotting the steady state solutions against a model parameter and using annotation to represent stability of solutions we can obtain a bifurcation diagram.

As an exercise perform a qualitative analysis of the Malthusian model.

4.3 The logistic growth model

4.3.1 Model development

Let $N = N(t)$. The logistic model, due to Verhulst, takes the form

$$\frac{dN}{dt} = rN(t)\left(1 - \frac{N(t)}{K}\right), \tag{4.2}$$

where r is the linear growth rate and K is carrying capacity. We consider both $r, K \in \Re^+$.

Questions to ask of such a model are: what type of biologically realistic solutions does it possess? Are there steady-states? If so, are they stable or unstable? Are there bifurcations in solutions?

4.3.1.1 Numerical solutions

In Figure **??** we present numerical solutions of equation using different initial conditions. Note the limiting behaviour of solutions as $t \to \infty$. In Figure **??** it is clear that even though some solutions are initialised at $N_0 = 0.1$, much closer to $N^* = 0$ than $N^* = K$, they tend to the limit $N = K$. Why do solutions not tend to $N^* = 0$?

4.3.1.2 Dimensional analysis and nondimensionalisation

N represents the population density and has units of one over area (say $1/m^2$) and t has units of time (say, seconds, s). Hence the left-hand side of Equation 4.2 has units of $1/(m^2 s)$. The first term on the right-hand side of Equation 4.2 is rN. N has units $1/m^2$ hence the parameter r must have units of $1/s$ for dimensional consistency. This is consistent as r represents the linear growth rate.

The second term has the form rN^2/K. Given the chosen units for r and N, the parameter K must have dimensions $1/m^2$. Again, this is consistent as K is a carrying capacity (i.e. it has units of population density).

We define the nondimensionalised variables

$$n = \frac{N}{\tilde{N}} \qquad \tau = \frac{t}{\tilde{T}}$$

where $\tilde{N}$ and $\tilde{T}$ are constants that have units of population density and time, respectively. Hence Equation 4.2 transforms, upon change of variables, to

$$\frac{\tilde{N}}{\tilde{T}}\frac{dn}{d\tau} = r\tilde{N}n(1 - \frac{n\tilde{N}}{K}).$$

In the case of the logistic equation there is only one time scale and density scale in the problem, hence we choose

$$\tilde{T} = \frac{1}{r} \quad and \quad \tilde{N} = K$$

and the dimensionless model is

$$\frac{dn}{d\tau} = n(1-n) \tag{4.3}$$

Note that we can retrieve the original equation by rescaling and calculating $N = \tilde{N}n$ and $t = \tilde{T}\tau$.

4.3.2 Steady states and linear stability

Steady states satisfy

$$n^*(1-n^*) = 0.$$

Hence

$$n^* = 0, \quad n^* = 1.$$

To determine linear stability we compute

$$H'(n) = (1-2n).$$

When $n = n^* = 0$ we obtain

$$H'(n) = 1.$$

Hence the origin is an unstable steady-state.

At the steady-state $n^* = 1$

$$H'(n^*) = -1$$

hence $n^* = 1$ is linearly stable.

Note that the linear stability analysis can explain the observations regarding the numeric solutions presented in Figure **??**.

4.3.3 Graphical analysis

In Figure Figure 4.1 we plot the right-hand side of Equation 4.3. We can qualitatively describe model solutions by considering the arrow along the x axis. Suppose we consider an initial condition with $0 < n_0 < 1$. Using the graph of $H(n)$, $dn/d\tau$ is positive, hence n increases as a function of time until $n(\tau) \to 1$.

```
import numpy as np
import matplotlib.pyplot as plt
N_max=2.1
```

```
K=2
r=0.2
N_vec=np.linspace(0,N_max,100)

rhs=r*N_vec*(1-N_vec/K)
fig, ax = plt.subplots(1)

ax.plot(N_vec, rhs)
plt.xlabel('$N$')
plt.ylabel('$H(N)$')
plt.show()
```

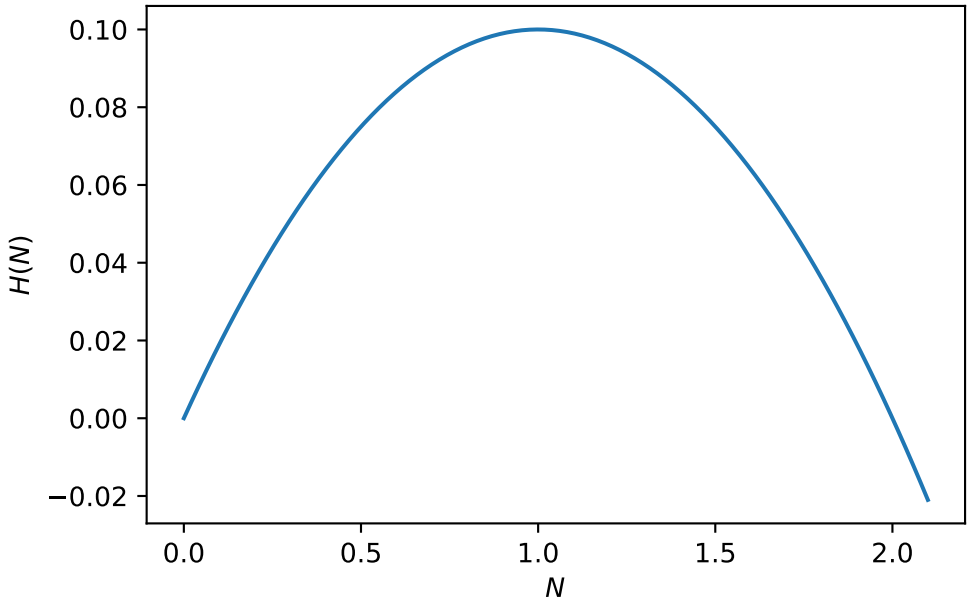

Figure 4.1: Right-hand side of the logistic ODE

4.3.4 An exact solution

Separation of variables yields

$$\int \frac{dN}{N(1-\frac{N}{K})} = r \int dt.$$

Using partial fractions

$$\int \frac{dN}{N} + \frac{1}{K} \int \frac{dN}{1-\frac{N}{K}} = r \int dt.$$

Integration yields

$$\ln N - \ln\left(1 - \frac{N}{K}\right) = \ln \frac{N}{1 - \frac{N}{K}} = rt + C.$$

Hence

$$N = \frac{De^{rt}}{1 + \frac{D}{K}e^{rt}}$$

Given an initial condition $N(0) = N_0$, we obtain

$$N(t) = \frac{N_0 K e^{rt}}{K + N_0(e^{rt} - 1)}$$

4.3.4.1 Qualitative analysis of the exact solution

As $t \to \infty$, $N \to K$. At $t = 0$, $N = N_0$ and that for small $N_0 \ll K$ the initial growth phase is exponential, i.e.

$$N(t) \sim N_0 e^{rt} \quad N_0 \ll K, t \ll \frac{1}{r}.$$

Note that in almost all the models that we will consider the above method is not an usually an option as the ODE is not explicitly integrable.

4.4 The spruce budworm model

The spruce budworm is a destructive and widely distributed forest defoliator in North America. Massive outbreaks occur periodically and can destroy large quantities of valuable spruce and fir. To understand the outbreak behaviour and develop and management strategies, a series of mathematical models have been developed, beginning with Ludwig et al. (1978). The goal of the models is to explain the qualitative pattern of sudden outbreaks and then a sudden collapse.

4.4.1 Model development

Letting $N(t)$ represent the population size at time t, it is assumed that budworm exhibits logistic growth and is subject to predation at rate $p(N)$. A governing ordinary differential equation is given by

$$\frac{dN}{dt} = r_B N\left(1 - \frac{N}{K_B}\right) - p(N), \tag{4.4}$$

where

$$p(N) = \frac{BN^2}{A^2 + N^2},$$

r_B is the linear growth rate, K_B is the carrying capacity, B is the maximum rate of predation and A is a measure of budworm population where predation switches on (specifically, A represents the budworm density at which predation is half its maximum value).

It is informative to graph the predation term

$$p(N) = \frac{BN^2}{A^2 + N^2},$$

and annotate the parameters A and B.

There is a root at $N = 0$. In the limit $N \to \infty$, $p \to B$. Note that $N = A$, $p = A/2$. The derivative is

$$p'(N) = \frac{2BN}{A^2 + N^2} - \frac{2BN^3}{(A^2 + N^2)^2} = \frac{2BA^2N}{(A^2 + N^2)^2}$$

Thus there is a turning point at $N = 0$ and $p' > 0 \forall N > 0$.

4.4.2 Nondimensionalisation

Introducing the (as yet unspecified) dimensional scalings $\tilde{N}$ and $\tilde{T}$, the model is nondimensionalised as follows

$$n = \frac{N}{\tilde{N}} \qquad \tau = \frac{t}{\tilde{T}}.$$

Changing variables in Equation 4.4 yields

$$\frac{\tilde{N}}{\tilde{T}}\frac{dn}{d\tau} = r_B\tilde{N}n\left(1 - \frac{\tilde{N}n}{K_B}\right) - \frac{B\tilde{N}^2n^2}{A^2 + \tilde{N}^2n^2}.$$

After some tidying

$$\frac{dn}{d\tau} = r_B\tilde{T}n\left(1 - \frac{\tilde{N}n}{K_B}\right) - \frac{B\tilde{N}\tilde{T}}{A^2}\frac{n^2}{1 + \frac{\tilde{N}^2}{A^2}n^2}.$$

A natural scale for cell density in the model is given by the parameter A, as it determines the density of budworm at which predation is half its maximal value. Hence we choose the scaling on the budworm density

$$\tilde{N} = A.$$

Similarly, a natural time scale for the model is given by

$$\tilde{T} = A/B.$$

Substituting for $\tilde{N}$ and $\tilde{T}$ yields

$$\frac{dn}{d\tau} = rn\left(1 - \frac{n}{q}\right) - \frac{n^2}{1+n^2} = H(n), \tag{4.5}$$

where we define the nondimensional parameters

$$r = \frac{r_B A}{B} \quad \text{and} \quad q = \frac{K_B}{A}.$$

Note that Equation 4.5 has two nondimensional parameters and all variables are dimensionless. See Figure 4.2 for a plot of right-hand side of equation Equation 4.5. What kind of behaviours do you expect to see from the model?

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

N_max=1.1
N_0=0.1
q=12
r_1=0.22
r_2=0.42
r_3=0.72

n_vec=np.linspace(0,N_max,100)

def rhssprucebudworm_model(x,t,r,q):

  rhs=r*x*(1-x/q) - x**2/(1+x**2)
  return rhs
def rhssprucebudworm_model_f(x,t,r,q):

  f=r*x*(1-x/q)
  return f
def rhssprucebudworm_model_g(x,t,r,q):

  g=x**2/(1+x**2)
  return g

rhs1=rhssprucebudworm_model(n_vec,0,r_1,q)
rhs2=rhssprucebudworm_model(n_vec,0,r_2,q)
rhs3=rhssprucebudworm_model(n_vec,0,r_3,q)
```

```
fig, ax = plt.subplots(1)

ax.plot(n_vec, rhs1,n_vec, rhs2,n_vec, rhs3)
plt.xlabel('$N$')
plt.ylabel('$H(N)$')
plt.grid(True)
plt.show()
```

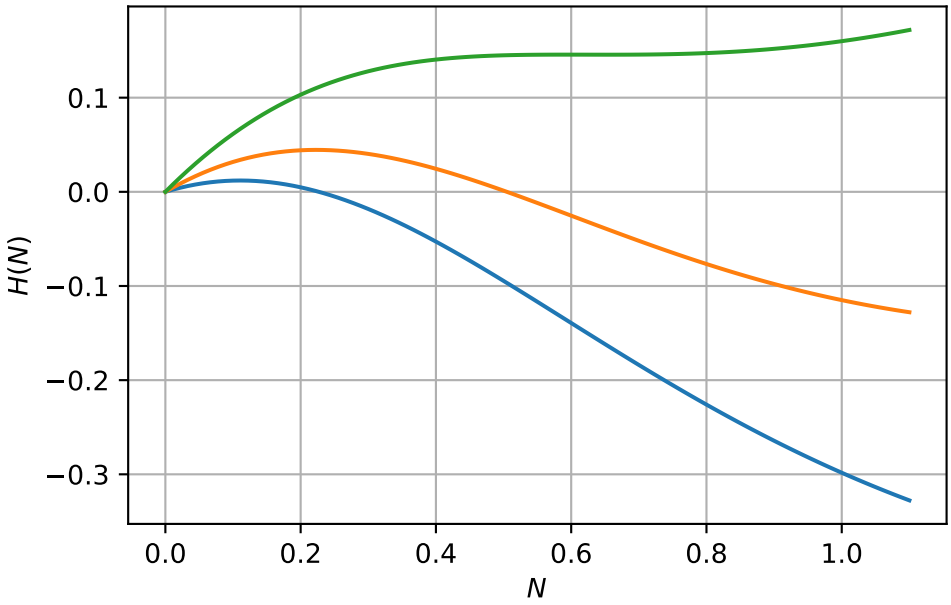

Figure 4.2: RHS of spr. budworm model

4.4.3 Numerical solutions

In Figure Figure 4.3 we plot some numerical solutions of Equation 4.5 at different values of the parameter r.

Numerical solutions of Equation 4.5 indicate that there is a single stable steady state when r is both small and large but for intermediate values of r there are two stable steady states.

Our goal is to analyse the model and understand why different parameter values yield these strikingly different model behaviours.

```
t = np.linspace(0, 100, 101)
sol1 = odeint(rhssprucebudworm_model, N_0, t, args=(r_1, q))
sol2 = odeint(rhssprucebudworm_model, N_0, t, args=(r_2, q))
sol3 = odeint(rhssprucebudworm_model, N_0, t, args=(r_3, q))

plt.plot(t, sol1, 'b', label='r=' +str(r_1))
plt.plot(t, sol2, 'r', label='r='+str(r_2))
plt.plot(t, sol3, 'm', label='r='+str(r_3))

plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

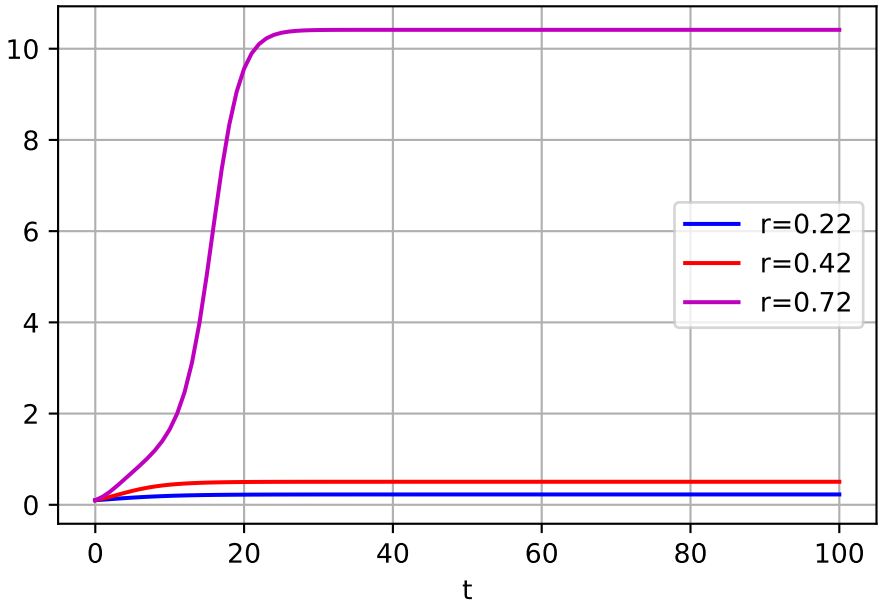

Figure 4.3: Numerical solution of spr. budworm model

4.4.4 Steady state analysis

Letting n^* represent steady states of Equation 4.5 yields

$$rn^*(1-\frac{n^*}{q})-\frac{n^{*2}}{1+n^{*2}}=0.$$

Hence either

$$n^* = 0,$$

or n^* satisfies the cubic equation

$$r\left(1 - \frac{n^*}{q}\right) - \frac{n^*}{1 + n^{*2}} = 0.$$

Explicit solutions to such a cubic can be immediately written down but they are cumbersome to work with. We proceed using a graphical/qualitative approach.

Define

$$f(n^*) = r\left(1 - \frac{n^*}{q}\right) \quad \text{and} \quad g(n^*) = \frac{n^*}{1 + n^{*2}}. \tag{4.6}$$

Roots occur for values of n^* that satisfy $f = g$.

In Figure 4.4 (a) we fix the parameter $q = 10$ and consider model behaviour as a function of the parameter r. When $r \gg 1$ there is a nonzero steady-state corresponding to $n^* \gg 1$. When $r \ll 1$ there is a nonzero steady-state corresponding to $n^* \ll 1$. In the intermediate case there can be three intersection points.

We can use the curve sketching techniques form Tutorial Sheet 1 to sketch f and g.

f is linear. There is a root at $n^* = q$. The derivative is $-r/q$. $f(0)$=r. g has a unique root at $n^* = 0$. The derivative is

$$g' = \frac{1 - n^{*2}}{(1 + n^*)^2}.$$

There is a turning point at $n^* = 1$. Here $g = 1/2$. As $n^* \to \infty$, $g \to 0$. $f'(0) = 1$.

4.4.5 Linear stability analysis

The linear stability of the model is determined by the quantity

$$H'(n) = r(1 - \frac{2n}{q}) - \frac{2n}{1 + n^2} + \frac{2n^3}{(1 + n^2)^2}.$$

Hence at the steady state $n^* = 0$

$$H'(0) = r$$

and the steady state is linearly unstable.

Given the nonzero steady states have not been calculated explicitly, we proceed using graphical analysis of stability. In Figure 4.4 (b) we plot the right-hand side of Equation 4.5 against n and examine the cases of large, small and intermediate r for a given value of q.

When r is both large and small the nonzero steady state is stable (the derivative at the roots is negative). In the case where three biologically relevant roots exist, the intermediate root is unstable.

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

N_max=13
N_0=0.1
q=12
r_1=0.2
r_2=0.5
r_3=0.8

n_vec=np.linspace(0,N_max,100)


def rhssprucebudworm_model_f(x,t,r,q):

  f=r*x*(1-x/q)
  return f
def rhssprucebudworm_model_g(x,t,r,q):

  g=x**2/(1+x**2)
  return g

f_1=rhssprucebudworm_model_f(n_vec,0,r_1,q)
f_2=rhssprucebudworm_model_f(n_vec,0,r_2,q)
f_3=rhssprucebudworm_model_f(n_vec,0,r_3,q)
g=rhssprucebudworm_model_g(n_vec,0,r_1,q)

h_1=rhssprucebudworm_model(n_vec,0,r_1,q)
h_2=rhssprucebudworm_model(n_vec,0,r_2,q)
h_3=rhssprucebudworm_model(n_vec,0,r_3,q)

fig, ax = plt.subplots(1,2)

ax[0].plot(n_vec, f_1,n_vec, f_2,n_vec, f_3,n_vec,g)
plt.xlabel('$n$')
plt.ylabel('$f,g$')
```

```
plt.grid(True)
ax[1].plot(n_vec, h_1,n_vec, h_2,n_vec, h_3)
plt.xlabel('$n$')
plt.ylabel('$h$')
plt.grid(True)
plt.show()
```

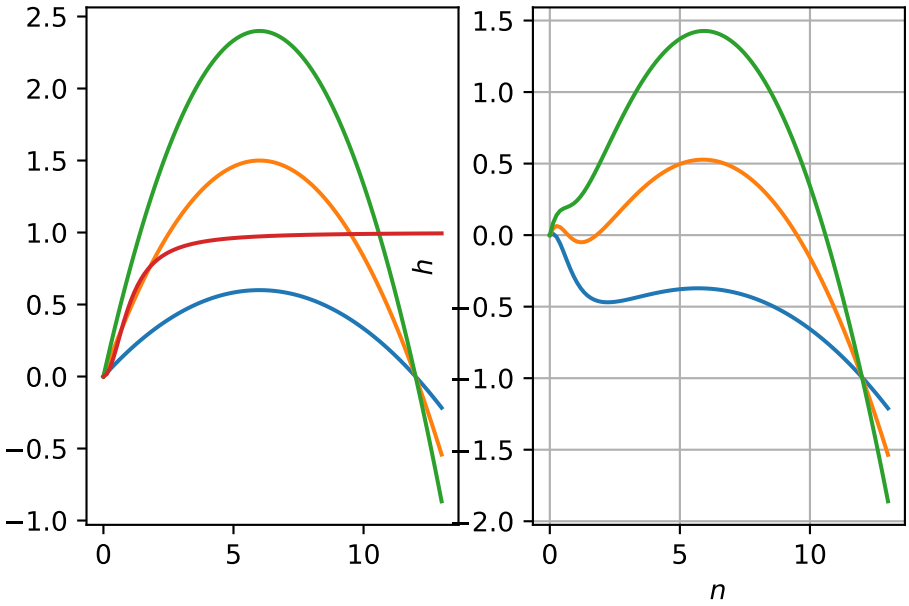

Figure 4.4: RHS of spr. budworm model

4.4.6 Bifurcation analysis

The goal is to identify boundaries of rq parameter space where the stability changes occur and/or the number of steady states changes.

We can define points in rq parameter space where bifurcations arise by seeking values of n^* that satisfy

$$f(n^*) = g(n^*) \qquad f'(n^*) = g'(n^*).$$

The first of these equations yields

$$r(1 - \frac{n^*}{q}) = \frac{n^*}{1 + n^{*2}}, \tag{4.7}$$

and the latter yields

$$-\frac{r}{q} = \frac{1}{1+n^{*2}} - \frac{2n^{*2}}{(1+n^*)^2} = \frac{1-n^{*2}}{(1+n^{*2})^2}. \tag{4.8}$$

Hence

$$\frac{r}{q} = \frac{n^{*2}-1}{(1+n^{*2})^2}.$$

Substituting for r/q in the first equation yields

$$r - \frac{n^{*2}-1}{(1+n^{*2})^2} n^* = \frac{n^*}{1+n^{*2}},$$

which can be written in the form

$$r = \frac{2n^{*3}}{(1+n^{*2})^2}.$$

Substituting for r in Equation 4.7 yields

$$\frac{2n^{*3}}{(1+n^{*2})^2} = \frac{2n^{*3}}{(1+n^{*2})^2}\frac{n^*}{q} + \frac{n^*}{1+n^{*2}}$$

which, after some algebra, yields

$$q = \frac{2n^{*3}}{n^{*2}-1}. \tag{4.9}$$

Hence a set of points that define bifurcations where three steady states transform to a single steady state are given in parametric form in qr parameter space by

$$\left(\frac{2n^{*3}}{n^{*2}-1}, \frac{2n^{*3}}{(1+n^{*2})^2}\right) \qquad n^* > 1.$$

```
def Computerq(x):
  r_n_star=2*x**3/(x**2-1)
  q_n_star=2*x**3/(1+x**2)**2
  return  r_n_star,q_n_star

n_s_1=np.linspace(1,np.sqrt(3),100)
n_s_2=np.linspace(np.sqrt(3),200,100)


r_1,q_1=Computerq(n_s_1)
r_2,q_2=Computerq(n_s_2)
```

```
fig, ax = plt.subplots(1)

ax.plot(r_1, q_1,r_2, q_2)
plt.xlabel('$r$')
plt.ylabel('$q$')
ax.set_xlim([0, 100])
plt.grid(True)

plt.show()
```

```
/var/folders/m_/vc0kz_0x6ls5n4qnksq052jw0000gp/T/ipykernel_70288/3415590409.py:2: RuntimeWar

divide by zero encountered in divide
```

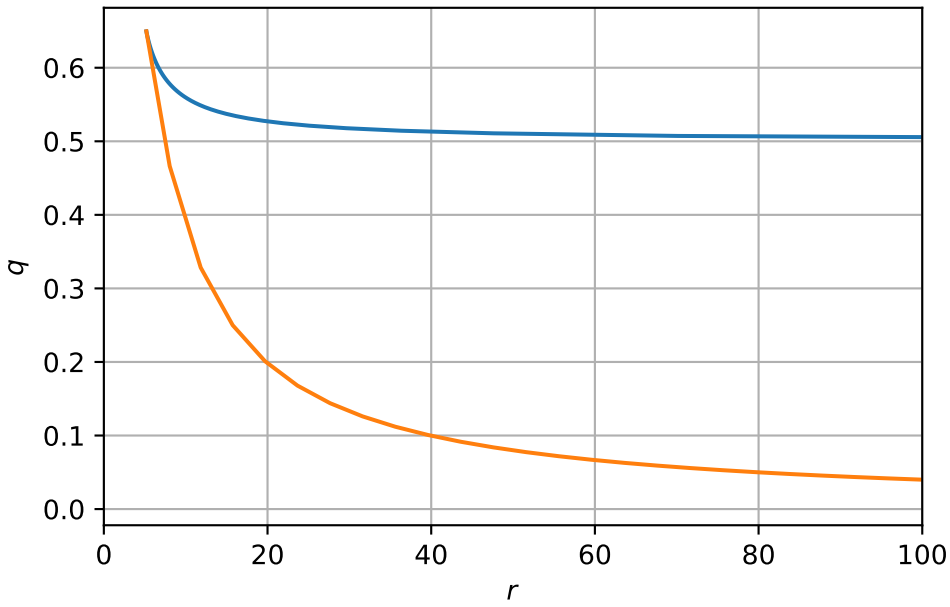

Figure 4.5: Bifurcations in the rq plane

By varying values of n^* in Figure 4.5 we plot a region of instability. Note for example that $q \to \infty$ as $n^* \to 1$ and as $n^* \to \infty$. Note also that in these limits r must take the value $1/2$ and 0, respectively.

We can show that the cusp in Figure 4.5 is given by

$$(q, r) = \left(3^{\frac{3}{2}}, \left(\frac{\sqrt{3}}{2}\right)^3\right).$$

Note that r is a decreasing function of q for $1 < n^* < \sqrt{3}$ but that r is an increasing function of n^* for $\sqrt{3} < n^* < \infty$. This can be shown by finding the turning points of r w.r.t n^*, i.e. Given that

$$r = \frac{2{n^*}^3}{(1 + {n^*}^2)^2},$$

differentiation with respect to n^* yields

$$\frac{dr}{dn^*} = \frac{6{n^*}^2}{(1 + n^*)^2} - \frac{8{n^*}^4}{(1 + n^*)^3}.$$

The turning point satisfies

$$\frac{6{n^*}^2}{(1 + n^*)^2} - \frac{8{n^*}^4}{(1 + n^*)^3} = 0$$

Solving for n^* yields

$$6(1 + {n^*}^2) - 8{n^*}^2 = 0$$

Hence

$$n^* = \sqrt{3}.$$

Thus there is a turning point that minimises r at $n^* = \sqrt{3}$.

Substitution for this value of n^* yields

$$(q, r) = \left(3^{\frac{3}{2}}, \left(\frac{\sqrt{3}}{2}\right)^3\right).$$

See Figure 4.5.

4.4.7 Hysteresis

Finally, rearranging the steady-state equation

$$r(1 - \frac{n^*}{q}) = \frac{n^*}{1 + {n^*}^2},$$

we obtain

$$r = \frac{n^*}{(1 + {n^*}^2)(1 - \frac{n^*}{q})}. \tag{4.10}$$

Considering $n^* < q$ we compute r; plotting n^* against r yields the bifurcation curve presented in Figure 4.6.

```
def Computern(x,q):
  r=x/((1+x**2)*(1-x/q))
  return  r


q=12
n_s_1=np.linspace(1,np.sqrt(3),100)
n_s_2=np.linspace(np.sqrt(3),0.99*q,100)
n_s_3=np.linspace(0,np.sqrt(3),100)



r_1=Computern(n_s_1,q)
r_2=Computern(n_s_2,q)
r_3=Computern(n_s_3,q)

fig, ax = plt.subplots(1)

ax.plot(r_1, n_s_1,r_2, n_s_2,r_3,n_s_3)
plt.xlabel('$r$')
plt.ylabel('$n^*$')
ax.set_xlim([0, 2])
plt.grid(True)

plt.show()
```

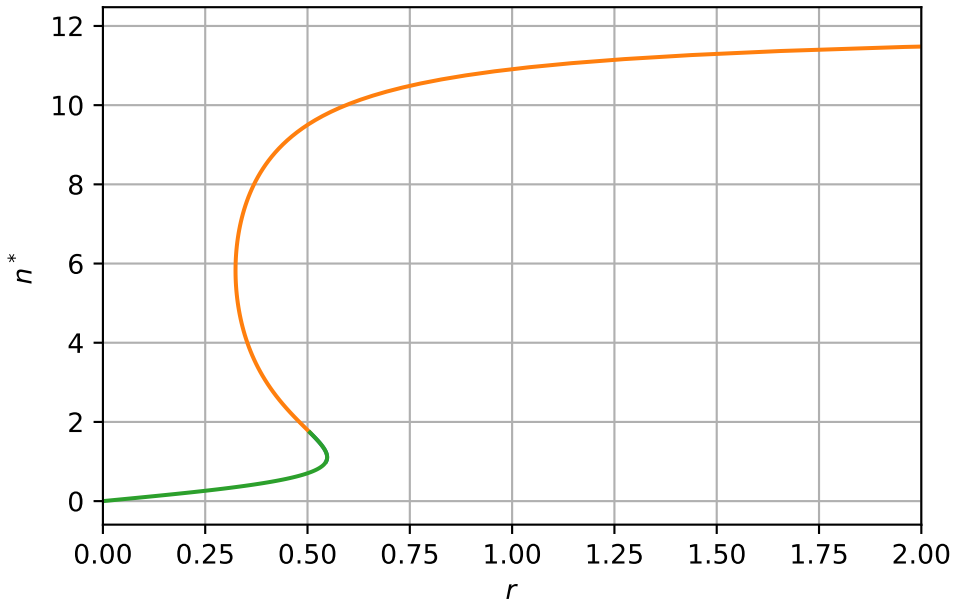

Figure 4.6: Bifurcations in the rq plane

A system exhibiting hysteresis shows a response to an increases in a control parameter that is not exactly reversed when the parameter is decreased. We can show that the spruce budworm model exhibits hysteresis by considering the argument below.

Suppose r is initially small ($r < r_1$). There is only one steady-state and any initial condition will converge towards it.

Suppose we increase the value of the parameter r. There will be a critical value of r ($r = r_1$)where a second stable steady-state arises and the model enters the bistable regime, where there are two possible stable steady-states. Given the system system was originally in the first stable steady state, it will remain there.

Suppose we continue to increase r. Eventually we reach critical value of r ($r = r_2$) where the first stable steady state is lost and there is again only one stable steady-state.

Suppose we now decrease the parameter r below the threshold $r = r_2$. The system again enters the bistable regime but as the second solution is stable, it remains the solution.

Suppose we continue to decrease r until eventually $r < r_1$. We return to the case where the system has only a single stable steady state.

4.5 Harvesting

By introducing terms that represent harvesting, population models can be used to investigate management strategies for resource management. The modelling problem is to maximise the sustained yield.

Consider a model for logistic growth supplemented with a harvesting term

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - EN.$$

The term EN is the harvesting yield per unit time and the constant E represents the harvesting effort.

Steady states satisfy

$$rN^*\left(1 - \frac{N^*}{K}\right) - EN^* = 0$$

Thus either $N^* = 0$ or

$$\left(1 - \frac{N^*}{K}\right) - E = 0$$

Hence

$$N^* = K(1 - \frac{E}{r}).$$

Note that this is positive only if

$$E < r.$$

Thus if the harvesting rate exceeds the linear growth rate the only steady state corresponds to extinction.

The yield is

$$Y(E) = EN^* = EK(1 - \frac{E}{r})$$

To maximise the yields we differentiate with respect to E.

$$\frac{dY}{dE} = K - 2\frac{EK}{r}.$$

Thus the maximum yield is identified by setting

$$\frac{dY}{dE} = K - 2\frac{EK}{r} = 0.$$

THe maximum occurs at harvesting rate

$$E_c = \frac{r}{2}.$$

The maximum yield is

$$E^* = \frac{rK}{4}.$$

We identify the time scale over which stocks return to steady state by linearising about the steady state $N^* = K(1 - E/r)$, hence

$$\frac{d\hat{N}}{dt} = (E - r)\hat{N}.$$

Hence the recovery time scale is

$$T_R(E) = O(\frac{1}{r - E}).$$

As the harvesting rate approaches the linear growth rate r, not only does the steady state population tend to zero but the timescale for stocks to recover tends to infinity. }

4.6 Delay differential equation models

Consider a model of the form

$$\frac{dN}{dt} = H(N(t), N(t - T)),$$

such that the right-hand side now depends on the value N not just at time t but also at some delay time $t - T$. We now need to prescribe initial conditions of the form

$$N(t) = f(t), \quad -T < t \leq 0.$$

Consider a simple case of a linear model

$$\frac{dN}{dt} = -N(t - T).$$

A steady state, N^*, is defined such that $N(t) = N(t - T) = N^*$. Hence

$$0 = -kN^*.$$

The only steady state is $N^* = 0$.

To compute the linear stability we consider solution of the linearised equation of the form

$$N(t) = e^{\lambda t}.$$

Hence

$$N(t - T) = e^{\lambda(t-T)}.$$

Substitution yields

$$\lambda e^{\lambda t} = -e^{\lambda(t-T)}.$$

Hence the eigenvalue, λ, satisfies a transcendental equation

$$\lambda = -e^{\lambda - T}.$$

This equation has no solutions for $\lambda \in \mathfrak{R}$.

Consider solutions $\lambda \in \mathbb{C}$. Let $\lambda = \mu + i\omega$. Substitution yields

$$\mu + i\omega = -e^{-\mu T}(\cos(\omega T) - i\sin(\omega T)).$$

Hence

$$\begin{aligned} \mu &= -\cos(\omega T)e^{-\mu T} \\ \omega &= \sin(\omega T)e^{-\mu T}. \end{aligned}$$

For linear stability we require that $\mu < 0$. This implies that

$$-\frac{\pi}{2} < \omega T < \frac{\pi}{2}.$$

Consider

$$\frac{e^{\mu T}}{T} = \frac{\omega e^{\mu T}}{\omega T} = \frac{\sin(\omega T)}{\omega T}$$

Noting that

$$\frac{\sin z}{z} > \frac{1}{\frac{\pi}{2}}$$

yields

$$\frac{\sin(\omega T)}{\omega T} > \frac{2}{\pi}.$$

Hence

$$\frac{e^{\mu T}}{T} > \frac{2}{\pi}.$$

Rearranging

$$T < \frac{\pi}{2} e^{\mu T}.$$

For linear stability $\mu < 0$. Hence a necessary condition for linear stability is that

$$T < \frac{\pi}{2}$$

5 Continuous time, multi species

In this chapter we consider models of interacitng populations.

Let $u = u(t)$ and $v = v(t)$ represent populations of two different species at continuous time t. Suppose that $u(t)$ and $v(t)$ satisfy the ordinary differential equations

$$\frac{du}{dt} = f(u, v),$$
$$\frac{dv}{dt} = g(u, v).$$

In this section we will consider particular forms for f and g that represent different types of inter species interaction. Before considering particular examples, we develop some general techniques for analysing equations of the form Equation 5.1.

5.1 General tools

5.1.1 Steady states

(u^*, v^*) is defined to be a steady state of Equation 5.1 if

$$f(u^*, v^*) = g(u^*, v^*) = 0.$$

Hence, by definition, the time derivatives of $u(t)$ and $v(t)$ are both zero at (u^*, v^*). As was the case for single species models, steady states are obtained by solving algebraic equations.

> **Note**
>
> Compute the steady states of the system of ODEs
>
> $$\frac{du}{dt} = 1 - u,$$
> $$\frac{dv}{dt} = 1 - uv - v.$$

Tip

Suppose (u^*, v^*) is a steady state. Hence

$$0 = 1 - u^*$$

and

$$0 = 1 - u^* v^* - v^*$$

The steady state is $(1, 1/2)$.

5.1.2 Linear stability analysis

Suppose that (u^*, v^*) is a steady state of equations Equation 5.1.

We consider a change of dependent variables such that

$$u(t) = u^* + \hat{u}(t) \quad \text{and} \quad v(t) = v^* + \hat{v}(t),$$

where $\hat{u}(t)$ and $\hat{v}(t)$ are perturbations about the steady state.

Rewriting equation Equation 5.1 in the transformed variables yields

$$\begin{aligned} \frac{d\hat{u}}{dt} &= f(u^* + \hat{u}, v^* + \hat{v}), \\ \frac{d\hat{v}}{dt} &= g(u^* + \hat{u}, v^* + \hat{v}). \end{aligned}$$

Making Taylor expansions about (u^*, v^*) yields the linearised equations

$$\begin{aligned} \frac{d\hat{u}}{dt} &= \frac{\partial f}{\partial u}_{|(u^*, v^*)} \hat{u} + \frac{\partial f}{\partial v}_{|(u^*, v^*)} \hat{v} + h.o.t, \\ \frac{d\hat{v}}{dt} &= \frac{\partial g}{\partial u}_{|(u^*, v^*)} \hat{u} + \frac{\partial g}{\partial v}_{|(u^*, v^*)} \hat{v} + h.o.t. \end{aligned}$$

Upon making the assumption that the perturbations about the steady state are small, the leading order terms are linear and higher order terms are neglected. Defining

$$\mathbf{w} = \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix},$$

yields the system of linear ODEs given by

$$\frac{d\mathbf{w}}{dt} = A\mathbf{w}, \tag{5.1}$$

where the matrix A, known as the Jacobian matrix, takes the form

$$A = \begin{pmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \\ \frac{\partial g}{\partial u} & \frac{\partial g}{\partial v} \end{pmatrix}_{(u^*,v^*)}.$$

The linear system arrived upon in Equation 5.1 was encountered previously in Differential Equations (MA31002). We summarise the important results here. Seeking a solution of Equation 5.1 of the form

$$\mathbf{w} = \mathbf{v}e^{\lambda t}$$

one obtains the characteristic equation

$$\lambda^2 - \lambda \mathrm{tr}(A) + \det(A) = 0,$$

which has solutions

$$\lambda = \frac{\mathrm{tr}A \pm \sqrt{\mathrm{tr}A^2 - 4\det A}}{2}.$$

Whilst a complete classification of the linear stability of a steady state can be obtained by explicitly calculating the eigenvalues, in many cases it is sufficient to deduce whether or not a steady state is stable or unstable. This can be achieved by calculating the determinant and trace of the Jacobian matrix ($\det(A)$ and $\mathrm{tr}(A)$, respectively) and referring to Figure **??**.

The different cases can be categorised as follows:

- $\det(A) < 0$ There is one positive and one negative real eigenvalue. Hence the steady state is a saddle which is unstable.
- $\det(A) > 0$ The steady state can be either stable or unstable, depending on $\mathrm{tr}(A)$ (and the real part of the eigenvalues).
  - If $\mathrm{tr}(A) > 0$, the steady state is unstable.
    - If $\mathrm{tr}(A)^2 > 4\det(A)$, it is an unstable node.
    - If $\mathrm{tr}(A)^2 < 4\det(A)$ it is an unstable spiral.
  - If $\mathrm{tr}(A) < 0$, the steady state is stable.
    - If $\mathrm{tr}(A)^2 > 4\det(A)$, it is an stable node.
    - If $\mathrm{tr}(A)^2 < 4\det(A)$, it is an stable spiral.

These different cases can be distinguished in the trace-determinant plane plotted in Figure ??.

Note that it can be shown rigorously that the linear stability of the nonlinear system is equivalent to that of the linearised system in all cases except when the steady state is a centre. In this case, nonlinear stability analysis is required to determine the stability of the nonlinear system.

Note

Deduce, by considering the form for the eigenvalues that, for example, the conditions $\det(A) > 0$, $\mathrm{tr}(A) > 0$ with $\mathrm{tr}(A)^2 < 4\det(A)$ imply that the steady state is an unstable spiral.

Tip

The eigenvalues are given by

$$\lambda = \frac{\mathrm{tr}A \pm \sqrt{\mathrm{tr}A^2 - 4\det A}}{2}$$

The conditiondet$(A) > 0$ excludes the case that the eigenvalues are real but have opposite signs (i.e. it cannot be a saddle point).
The condition $\mathrm{tr}(A) > 0$ implies that the real part of both eigenvalues are positive (i.e. the steady state is unstable).
The condition $\mathrm{tr}(A)^2 < 4\det(A)$ implies that the eigenvalue are complex. Hence

$$\lambda_{\pm} = \mu \pm i\omega$$

and the solution of the system can be written

$$e^{\lambda t} = e^{(\mu + i\omega)t} = e^{\mu t}e^{i\omega t}.$$

Thus the magnitude of perturbation grows but it oscillates about the steady state. Hence the steady state is an unstable spiral.

Note

Compute the Jacobian matrix for the system of ODEs

$$\frac{du}{dt} = 1 - u,$$
$$\frac{dv}{dt} = 1 - uv - v.$$

Evaluate the Jacobian matrix at the steady state and hence determine its linear stability.

Tip

The Jacobian is given by

$$A = \begin{pmatrix} -1 & 0 \\ -v & -u-1 \end{pmatrix}.$$

At (1,1/2)

$$A = \begin{pmatrix} -1 & 0 \\ -\frac{1}{2} & -2 \end{pmatrix}.$$

In this case

$$tr(A) = -3$$

and

$$\det(A) = 2$$

Hence the steady state is stable. As

$$tr(A)^2 - 4\det(A) = 9 - 8 = 1,$$

it is a stable node.

5.1.3 Nullclines

Nullclines of the equations Equation 5.1 are given by the curves

$$f(u, v) = 0$$

and

$$g(u, v) = 0,$$

respectively. Note that steady states arise at the intersection of nullclines.

Using the u nullclines we identify domains of the phase plane where $du/dt > 0$ and $du/dt < 0$.

Similarly, using the v nullcline we identify domains of the phase plane where $dv/dt > 0$ and $dv/dt < 0$.

The nullclines can be used to help identify confined sets and hence apply the Poincaire-Bendixson theorem.

Note

Sketch the nullclines of the system of ODEs

$$\frac{du}{dt} = 1 - u,$$
$$\frac{dv}{dt} = 1 - uv - v.$$

Tip

The u nullcline is $u = 1$. The v nullcline is $v = 1/(1 + u)$.

5.1.4 Poincaire-Bendixson theorem

Suppose that the system of equations

$$\frac{du}{dt} = f(u, v), \qquad \frac{dv}{dt} = g(u, v) \tag{5.2}$$

possesses a confined set (i.e. a bounded domain in the phase plane upon which the derivative field points into the domain) that contains an unstable node or spiral. Any trajectory cannot leave the confined set, nor can it tend to the unstable steady state. The Poincaire-Bendixson theorem states that as $t \to \infty$, the trajectory will tend towards a limit cycle.

5.1.5 Dulac criterion

Suppose D is a simply connected region in the plane and that there exists a function $B(x, y)$, continuously differentiable on D, such that the expression

$$\frac{\partial}{\partial u}(Bf) + \frac{\partial}{\partial v}(Bg)$$

is not identically zero and does not change sign in D. Then there are no closed orbits in D.

5.2 The Lotka-Voltera model

Suppose that $N(t)$ and $P(t)$ represent the prey and predator population densities at time t, respectively. Consider a model of the form

$$\frac{dN}{dt} = aN - bNP,$$
$$\frac{dP}{dt} = cNP - dP,$$

where a, b, c and d are positive constants.

5.2.1 Nondimensionalisation

Nondimensionalising with

$$n = \frac{N}{\tilde{N}}, \quad p = \frac{P}{\tilde{P}} \text{ and } \tau = \frac{t}{\tilde{T}},$$

yields, after changing variables,

$$\frac{dn}{d\tau} = \tilde{T}an - b\tilde{T}\tilde{P}np,$$
$$\frac{dp}{d\tau} = c\tilde{N}\tilde{T}np - d\tilde{T}p.$$

Choosing the dimensional scalings

$$\tilde{T} = 1/a, \quad \tilde{P} = a/b \quad \text{and} \quad \tilde{N} = d/c,$$

yields

$$\frac{dn}{d\tau} = n(1-p) = f(n,p),$$
$$\frac{dp}{d\tau} = \alpha p(n-1) = g(n,p),$$

where $\alpha = d/a$. Note that all variables and parameter are without dimensions.

5.2.2 Numerical solutions

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


alpha=1.5
def rhs_lv_model(x,t):
  rhs=np.zeros_like(x)
  n=x[0]
  p=x[1]
  dn_dt=n*(1-p)
  dp_dt=alpha*p*(n-1)
```

```
  rhs[0]=dn_dt
  rhs[1]=dp_dt
  return rhs

t = np.linspace(0, 10, 1000)

init_cond=[0.5,0.5]
alpha=2.0
sol1 = odeint(rhs_lv_model, init_cond,t)


n=sol1[:,0]
p=sol1[:,1]

plt.plot(t, n, 'b',t,p,'r')

plt.legend(['n','p'],loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

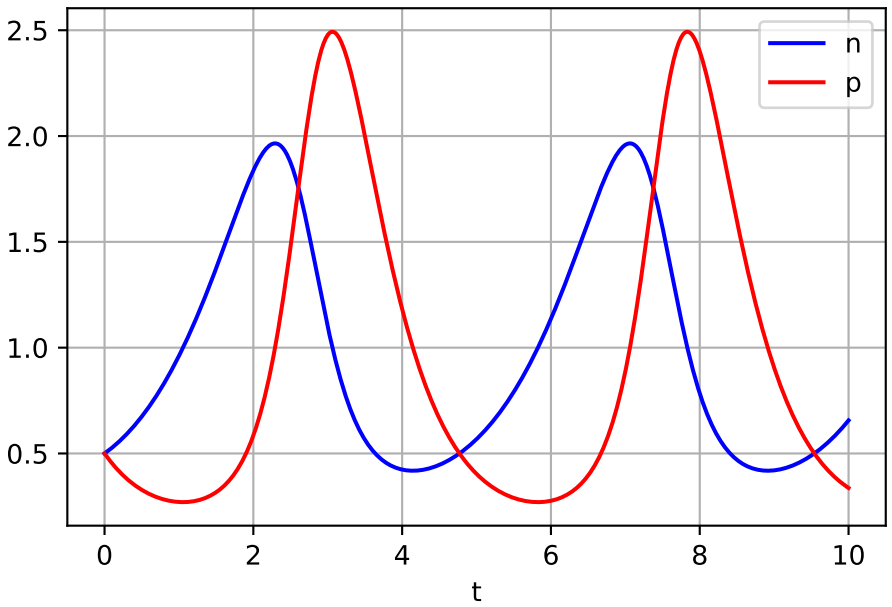

Figure 5.1: Numerical solution of Lotka-Volterra model

5.2.3 Steady states

The steady-states of equation Equation 5.3 are identified by seeking solutions of

$$f(n^*, p^*) = g(n^*, p^*) = 0.$$

Hence

$$n^*(1 - p^*) = 0 \qquad \alpha p^*(n^* - 1).$$

The first of these equations has solutions either

$$n^* = 0 \quad \text{or} \quad p^* = 1.$$

Substituting for $n^* = 0$ in the second equation yields $p^* = 0$. Hence one steady state is $(0, 0)$. Substituting for $p^* = 1$ in the second equation yields $n^* = 1$. Hence a second steady state is (1,1).

5.2.4 Linear stability

The linear stability of the steady states is described by the Jacobian matrix

$$A = \begin{pmatrix} \frac{\partial f}{\partial n} & \frac{\partial f}{\partial p} \\ \frac{\partial g}{\partial n} & \frac{\partial g}{\partial p} \end{pmatrix}_{(n^*, p^*)} = \begin{pmatrix} 1 - p & -n \\ \alpha p & \alpha(n - 1) \end{pmatrix}_{(n^*, p^*)}$$

Evaluating at (0,0) yields

$$A = \begin{pmatrix} 1 & 0 \\ 0 & -\alpha \end{pmatrix},$$

Hence the eigenvalues of A are 1 and $-\alpha$. As $\alpha > 0$ the origin is a saddle. The eigenvectors are $[1, 0]$ and $[0,1]$. $

At (1,1)

$$A = \begin{pmatrix} 0 & -1 \\ \alpha & 0 \end{pmatrix},$$

and the eigenvalues are $\pm i\sqrt{\alpha}$. Therefore the steady state at (1,1) is a centre.

5.2.5 Solutions in the phase plane

See Figure Figure 5.2 for solution trajectories plotted in the phase plane using different initial conditions. Note the expected saddle like behaviour when trajectories that are close to the origin. Furthermore, note that the different initial conditions result in distinct closed loops and that the inner-most loop, i.e. that closest to the steady-state (1,1), behaves like a centre, as expected given the linear stability analysis. However, far from the steady-state the solution trajectory deviates from the linearised model.

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


alpha=1.5
def rhs_lv_model(x,t):
  rhs=np.zeros_like(x)
  n=x[0]
  p=x[1]
  dn_dt=n*(1-p)
  dp_dt=alpha*p*(n-1)
  rhs[0]=dn_dt
  rhs[1]=dp_dt
  return rhs


t = np.linspace(0, 10, 1000)

init_cond1=[0.75,0.75]
init_cond2=[0.15,0.15]
init_cond3=[2.5,0.5]

alpha=2.0
sol1 = odeint(rhs_lv_model, init_cond1,t)
sol2 = odeint(rhs_lv_model, init_cond2,t)
sol3 = odeint(rhs_lv_model, init_cond3,t)

n=sol1[:,0]
p=sol1[:,1]

ss=[1,1]
```

```
plt.plot(sol1[:,0],sol1[:,1],sol2[:,0],sol2[:,1],sol3[:,0],sol3[:,1])
plt.plot(ss[0],ss[1],'k*')
plt.xlabel('$n$')
plt.ylabel('$p$')



plt.grid()
plt.show()
```

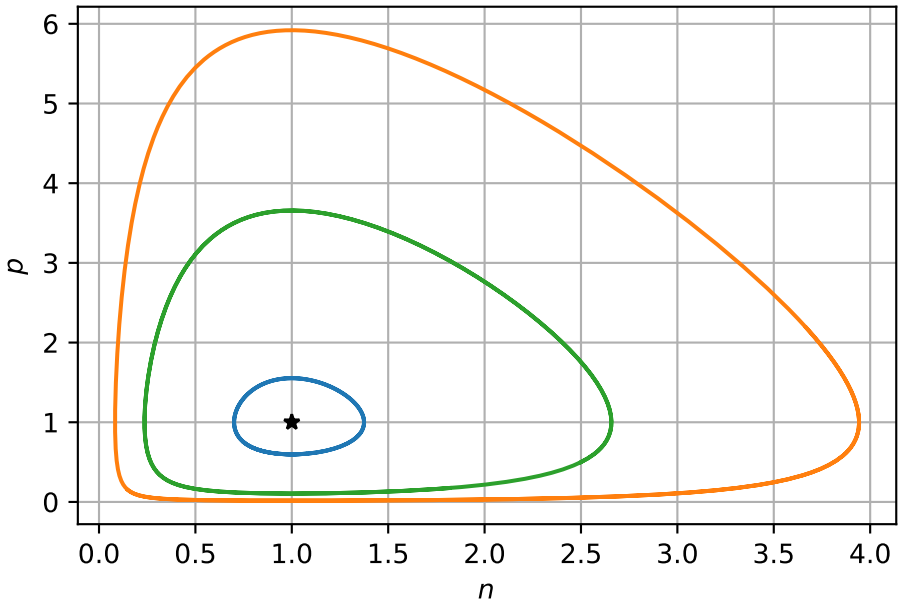

Figure 5.2: Numerical solution of Lotka-Volterra model

5.2.6 Direct integration

The Lotka-Volterra equations are a special case as they are integrable. Trajectories in the phase plane satisfy the differential equation

$$\frac{dp}{dn} = \frac{\alpha p(n-1)}{n(1-p)},$$

Using separation of variables

$$\int \frac{1-p}{p} dp = \alpha \int \frac{n-1}{n} dn.$$

Integration yields

$$\ln p - p = \alpha(n - \ln n) + H,$$

where H is a conserved quantity that this is determined by the initial conditions. As the equations take conservative form, the Lotka-Voltera model is said to be structurally unstable, as a small perturbation to the solution at a given point in the oscillatory cycle can result in large changes elsewhere in the cycle. For example, suppose the outermost limit cycle in Figure Figure 5.2 is perturbed by a small amount at the point (1,0.1) onto its nearest limit cycle. Later in the cycle these two trajectories deviate by a large amount.

5.3 Competition

In models of competition, two or more species compete for the same resource or in some way inhibit each other's growth. Letting $N_1(t)$ and $N_2(t)$ represent the population density of two species, we consider the ODEs

$$\frac{dN_1}{dt} = r_1 N_1 \left(1 - \frac{N_1}{K_1} - b_{12}\frac{N_2}{K_1}\right),$$
$$\frac{dN_2}{dt} = r_2 N_2 \left(1 - \frac{N_2}{K_2} - b_{21}\frac{N_1}{K_2}\right),$$

where r_1, r_2, K_1 and K_2 are positive constants. As before, the $r's$ are linear growth rates and the K's are carrying capacities. The parameters b_{12} and b_{21} measure the competitive effect of N_2 on N_1 and N_1 on N_2, respectively.

5.3.1 Nondimensionalisation

After nondimensionalising using the change of variables

$$n_1 = \frac{N_1}{K_1} \quad n_2 = \frac{N_2}{K_2} \quad \tau = \frac{t}{\frac{1}{r_1}},$$

we obtain the equations

$$\frac{dn_1}{d\tau} = n_1\left(1 - n_1 - a_{12}n_2\right) = f(n_1, n_2),$$
$$\frac{dn_2}{d\tau} = \rho n_2\left(1 - n_2 - a_{21}n_1\right) = g(n_1, n_2),$$

where

$$\rho = \frac{r_2}{r_1}, \quad a_{12} = b_{12}\frac{K_2}{K_1}, \quad a_{21} = b_{21}\frac{K_1}{K_2}.$$

5.3.2 Steady states

The steady states of Equation 5.3 are identified in the usual manner, i.e. by seeking $({n_1}^*, {n_2}^*)$ such that

$$f({n_1}^*, {n_2}^*) = g({n_1}^*, {n_2}^*) = 0.$$

The steady state equations are

$${n_1}^*\left(1 - {n_1}^* - a_{12}{n_2}^*\right) = 0 \qquad {n_2}^*\left(1 - {n_2}^* - a_{21}{n_1}^*\right) = 0.$$

The first equation has solution

$$n_1^* = 0$$

or

$$\left(1 - {n_1}^* - a_{12}{n_2}^*\right) \implies n_2 = \frac{1}{a_{12}}(1 - n_1^*).$$

Consider $n_1^* = 0$. Substitution in the second equation yields

$${n_2}^*\left(1 - {n_2}^*\right) = 0.$$

Hence either $n_2^* = 0$ or $n_2^* = 1$. Hence two steady states are $(0, 0)$ and $(0, 1)$. \ Now consider $n_2^* = \frac{1}{a_{12}}(1 - n_1^*)$ with $n_1^* \neq 0$. \ Substitution in the second steady state equation yields

$$\frac{1}{a_{12}}(1 - n_1^*)\left(1 - \frac{1}{a_{12}}(1 - n_1^*) - a_{21}{n_1}^*\right)$$

Hence either $n_1^* = 1$ or

$$\left(1 - \frac{1}{a_{12}}(1 - n_1^*) - a_{21}{n_1}^*\right) = 0 \implies n_1^* = \frac{1 - a_{12}}{1 - a_{12}a_{21}}.$$

In the case where $n_1^* = 1$, we find that $n_2^* = 0$. Hence the steady state is (1,0).

In the case where

$$n_1^* = \frac{1 - a_{12}}{1 - a_{12}a_{21}}$$

we find that

$$n_2^* = \frac{1 - a_{21}}{1 - a_{12}a_{21}}$$

Hence the steady state is

$$\left(\frac{1 - a_{12}}{1 - a_{12}a_{21}}, \frac{1 - a_{21}}{1 - a_{12}a_{21}} \right).$$

5.3.3 Nullclines

The nullclines for Equation 5.3 are straight lines given by

$$n_1 = 0 \qquad n_2 = \frac{1 - n_1}{a_{12}},$$

and

$$n_2 = 0 \qquad n_2 = 1 - a_{21}n_1.$$

Note that the steady states $(0, 0)$, $(1, 0)$ and $(0, 1)$ are always biologically relevant (i.e. independently of the parameter values for a_{12} and a_{21}).

However, the coexistence steady state is only biologically relevant if the nullclines intersect in the positive quadrant and this occurs only in certain regions of the model's parameter space.

In the cases where $a_{12}, a_{21} < 1$ and $a_{12}, a_{21} > 1$ there is a coexistence steady state (i.e. the nullclines intersect in the positive quadrant).

However, if $a_{21} < 1$ and $a_{12} > 1$ or $a_{12} < 1$ and $a_{21} > 1$ there is not a biologically relevant, coexistence steady state (i.e. the nullclines do not intersect in the positive quadrant).

Hence there are are four qualitatively different types of solution to consider.

5.3.4 Linear stability

The linear stability of the different steady states is determined by calculating the Jacobian matrix

$$A = \begin{pmatrix} \frac{\partial f}{\partial n_1} & \frac{\partial f}{\partial n_2} \\ \frac{\partial g}{\partial n_1} & \frac{\partial g}{\partial n_2} \end{pmatrix}_{(n_1{}^*, n_2{}^*)} = \begin{pmatrix} 1 - 2n_1 - a_{12}n_2 & -a_{12}n_1 \\ -\rho a_{21}n_2 & \rho(1 - 2n_2 - a_{21}n_1) \end{pmatrix}_{(n_1{}^*, n_2{}^*)}.$$

At (0,0)

$$A = \begin{pmatrix} 1 & 0 \\ 0 & \rho \end{pmatrix}.$$

Hence the eigenvalues of the Jacobian are 1 and ρ. As $\rho > 0$, the origin is therefore an unstable node (there are two real positive eigenvalues).

At (1,0)

$$A = \begin{pmatrix} -1 & -a_{12} \\ 0 & \rho(1 - a_{21}) \end{pmatrix}.$$

The trace and determinant are given by

$$\det A = \rho(a_{21} - 1) \quad \text{and} \quad \mathrm{tr}A = -1 + \rho(1 - a_{21}).$$

Hence if $a_{21} < 1$, $\det A < 0$ and (1,0) is a saddle point and thus unstable (see Figure **??**).

If $a_{21} > 1$, $\det A > 0$ and $\mathrm{tr}A < 0$. Hence (1,0) is a stable node.

Hence the parameter a_{21}, which describes how strongly Population 1 inhibits the growth rate of Population 2, determines whether or not the steady state representing extinction of Population 2 but not Population 1 is stable or not.

At (0,1)

$$A = \begin{pmatrix} 1 - a_{12} & 0 \\ -\rho a_{21} & -\rho \end{pmatrix}.$$

In this case if $a_{12} < 1$, $\det A < 0$ and (0,1) is a saddle point. If $a_{12} > 1$, $\det A > 0$ and $\mathrm{tr}A < 0$ and (0,1) is a stable node. Hence the parameter a_{12}, which describes how strongly Population 2 inhibits the growth rate of Population 1, determines whether or not the steady state representing extinction of Population 1 but not Population 2 is stable or not.

At the coexistence steady state, recall the steady state is

$$\left(\frac{1-a_{12}}{1-a_{12}a_{21}}, \frac{1-a_{21}}{1-a_{12}a_{21}}\right).$$

Note that this steady state is only biologically relevant in the cases $a_{21} < 1, a_{12} < 1$ or $a_{21} > 1, a_{12} > 1$. Evaluating the Jacobian yields

$$A = \frac{1}{1-a_{12}a_{21}} \begin{pmatrix} a_{12}-1 & -a_{12}(1-a_{12}) \\ -\rho a_{21}(1-a_{21}) & \rho(a_{21}-1) \end{pmatrix}.$$

The determinant and trace of the Jacobian are given by

$$\begin{aligned} \det A &= \rho\left((a_{12}-1)(a_{21}-1) - a_{12}a_{21}(1-a_{12})(1-a_{21})\right)\frac{1}{(1-a_{12}a_{21})^2}, \\ &= \rho\frac{(a_{12}-1)(a_{21}-1)}{(1-a_{12}a_{21})}, \end{aligned}$$

and

$$\mathrm{tr}A = \left(a_{12}-1+\rho(a_{21}-1)\right)\frac{1}{1-a_{12}a_{21}},$$

respectively. \stwriting{

Let's firstly consider the case where $a_{21} < 1$ and $a_{21} < 1$. This implies that $a_{21} - 1 < 0$ and $a_{12} - 1 < 0$, hence evaluating the signs of the different products yields

$$\det A = \rho(-)(-)(+) > 0$$

and

$$\mathrm{tr}A = \rho(-) + (-) < 0.$$

Therefore the coexistence steady state is a stable node or spiral.

In the case where $a_{21} > 1$ and $a_{12}>1$

$$\det A = \rho(+)(+)(-) < 0.$$

Hence the coexistence steady state is a saddle.

5.3.5 Phase portrait

See Figure Figure 5.4 for phase portraits of three of the four cases that we have considered. It is expected that you can sketch phase portraits. Key details to consider are the steady states and their linear stability. You should also sketch the nullclines and depict the sign of the derivatives in the phase plane on either side of the nullclines. You should also sketch one or more sample trajectories.

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


rho=1.5
a_12=0.5
a_21=0.5
def rhs_comp_model(x,t):
  rhs=np.zeros_like(x)
  n_1=x[0]
  n_2=x[1]
  dn_1_dt=n_1*(1-n_1)-a_12*n_1*n_2
  dn_2_dt=rho*(n_2*(1-n_2)-a_21*n_1*n_2)
  rhs[0]=dn_1_dt
  rhs[1]=dn_2_dt
  return rhs

t = np.linspace(0, 10, 1000)

init_cond=[0.5,0.5]
sol1 = odeint(rhs_comp_model, init_cond,t)


n_1=sol1[:,0]
n_2=sol1[:,1]

plt.plot(t, n_1, 'b',t,n_2,'r')

plt.legend(['$n_1$','$n_2$'],loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

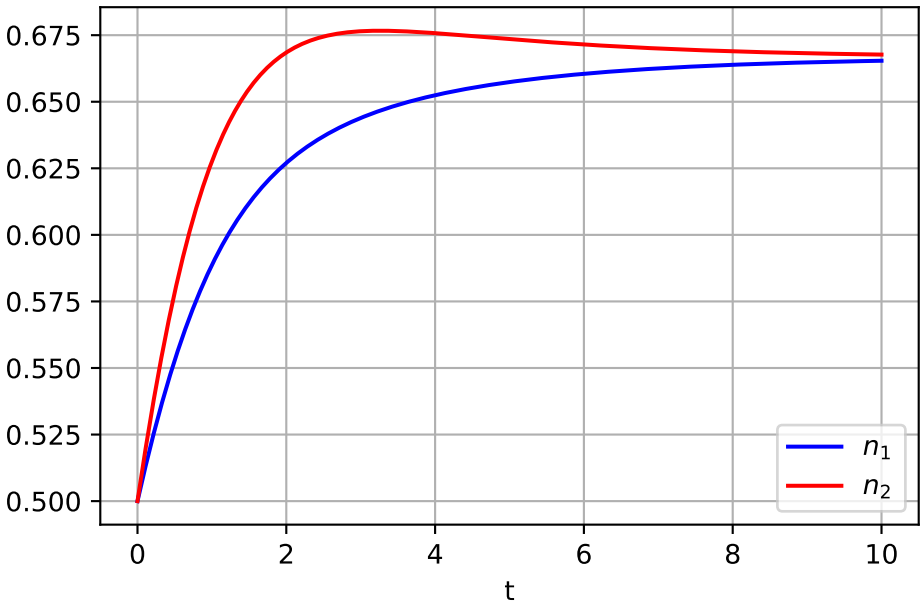

Figure 5.3: Numerical solutions of competition model

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



n_1_vec=np.linspace(0,5,100)

def ComputeCompetitionSol(a_12,a_21,n_1_vec):
    t = np.linspace(0, 10, 1000)

    init_cond1=[0.75,0.75]
    init_cond2=[0.15,0.15]
    init_cond3=[2.5,0.5]

    alpha=2.0
    sol1 = odeint(rhs_comp_model, init_cond1,t)
    sol2 = odeint(rhs_comp_model, init_cond2,t)
    sol3 = odeint(rhs_comp_model, init_cond3,t)
```

```
    num_steady_states=3
    fourth_ss_condition= ((a_12<1) & (a_21<1)) | ((a_12>1) & (a_21>1))
    if fourth_ss_condition==True:
        num_steady_states=4

    ss=np.zeros((num_steady_states,2),dtype=float)
    ss[0,:]=[0,0]
    ss[1,:]=[1,0]
    ss[2,:]=[0,1]
    if fourth_ss_condition==True:
        ss[3,:]=[(1-a_12)/(1-a_12*a_21),(1-a_21)/(1-a_12*a_21)]


    n1_ncline_1_n_1=[0,0]
    n1_ncline_1_n_2=[0,5]
    n1_ncline_2_n_2=1/a_12*(1-n_1_vec)
    n2_ncline_1_n_1=[0,5]
    n2_ncline_1_n_2=[0,0]
    n2_ncline_2_n_2=1-a_21*(n_1_vec)

    return sol1,sol2,sol3,ss,n1_ncline_1_n_1,n1_ncline_1_n_2,n1_ncline_2_n_2,n2_ncline_1_n


a_12=0.5
a_21=0.4
sol1,sol2,sol3,ss, n1_ncline_1_n_1,n1_ncline_1_n_2,n1_ncline_2_n_2,n2_ncline_1_n_1,n2_ncli

fig, ax = plt.subplots(1,2)
ax[0].plot(n1_ncline_1_n_1,n1_ncline_1_n_2,'k--')
ax[0].plot(n_1_vec,n1_ncline_2_n_2,'k--')
ax[0].plot(n2_ncline_1_n_1,n2_ncline_1_n_2,'r--')
ax[0].plot(n_1_vec,n2_ncline_2_n_2,'r--')

ax[0].plot(sol1[:,0],sol1[:,1],sol2[:,0],sol2[:,1],sol3[:,0],sol3[:,1])
ax[0].plot(ss[:,0],ss[:,1],'k*')
plt.xlabel('$n_1$')
plt.ylabel('$n_2$')

ax[0].set_xlim([-0.05,1.5])
ax[0].set_ylim([-0.05,1.5])
```

```
a_12=1.5
a_21=0.4
sol1,sol2,sol3,ss, n1_ncline_1_n_1,n1_ncline_1_n_2,n1_ncline_2_n_2,n2_ncline_1_n_1,n2_ncli
ax[1].plot(n1_ncline_1_n_1,n1_ncline_1_n_2,'k--')
ax[1].plot(n_1_vec,n1_ncline_2_n_2,'k--')
ax[1].plot(n2_ncline_1_n_1,n2_ncline_1_n_2,'r--')
ax[1].plot(n_1_vec,n2_ncline_2_n_2,'r--')

ax[1].plot(sol1[:,0],sol1[:,1],sol2[:,0],sol2[:,1],sol3[:,0],sol3[:,1])
ax[1].plot(ss[:,0],ss[:,1],'k*')
plt.xlabel('$n_1$')
plt.ylabel('$n_2$')

ax[1].set_xlim([-0.05,1.5])
ax[1].set_ylim([-0.05,1.5])


plt.grid()
plt.show()
```

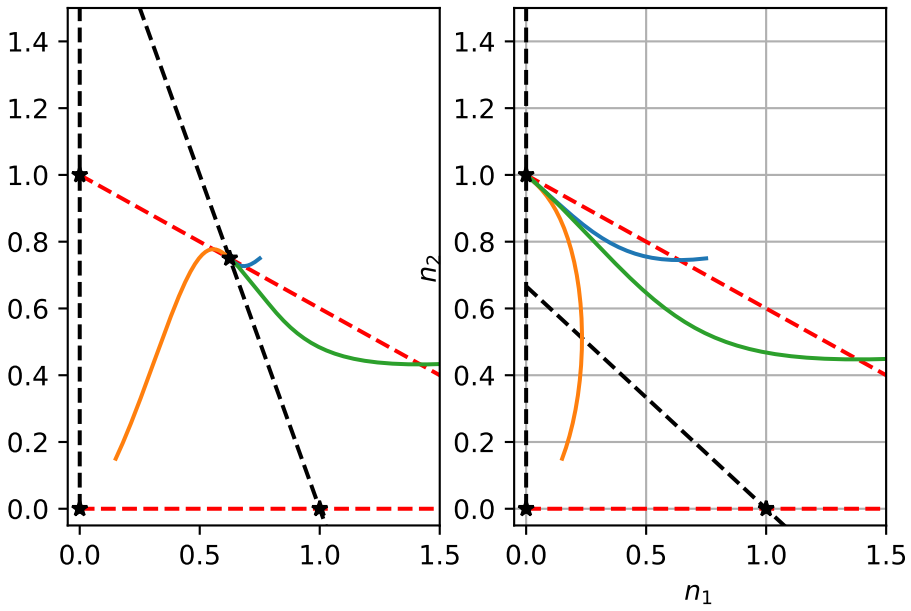

Figure 5.4: Numerical solution of the competition model

5.3.6 Insight

The model therefore has four qualitatively different behaviours that are described as follows: Consider the case where $a_{21} > 1$. This represents the case of Population 1 strongly competing with Population 2.

- If $a_{12} > 1$, Population 2 also strongly competes with Population 1. In this case, there are four biologically relevant steady states, two of which are stable (1,0) and (0,1). The coexistence steady state is a saddle and thus unstable. The model is bistable and the initial conditions determine whether solutions end up at (1,0) or (0,1) (see Figure **??** (c)). The biological interpretation of this solution is that one species will always win and completely outcompete the other. Even if the two populations are equal ($K_1 = K_2$ and $a_{21} = a_{12} > 1$), one species will always win and the other will become extinct.
- If $a_{12} < 1$, Population 2 weakly competes with Population 1. There is no coexistence steady state and the only stable steady state is (1,0). Hence Population 1 always wins and Population 2 always becomes extinct.

Now consider the case where $a_{21} < 1$. This represents the case of Population 1 weakly competing with Population 2.

- If $a_{12} > 1$, Population 2 strongly competes with Population 1. There is nonexistence steady state and the only stable steady state is (0,1). Hence Population 2 always wins and Population 1 always becomes extinct.
- If $a_{12} < 1$, Population 2 also weakly competes with Population 1. The coexistence steady state is stable and the steady states (1,0) and (0,1) are unstable.

5.4 Symbiosis/Mutualism

When the interactions between two species results in mutually benefit, it is known as mutualism or symbiosis. To study this behaviour, we consider a model of the form

$$\frac{dN_1}{dt} = r_1 N_1 \left(1 - \frac{N_1}{K_1} + b_{12}\frac{N_2}{K_1}\right),$$
$$\frac{dN_2}{dt} = r_2 N_2 \left(1 - \frac{N_2}{K_2} + b_{21}\frac{N_1}{K_2}\right).$$

Note the only difference with the competition model is that the sign of the interaction term has changed. Hence we will not work through all the details as the analysis is the same as before.

5.4.1 Nondimensionalisation

Using the same nondimensionalisation as the competition model, we obtain the nondimensional equations

$$\frac{dn_1}{d\tau} = n_1(1 - n_1 + a_{12}n_2) = f(n_1, n_2),$$
$$\frac{dn_2}{d\tau} = \rho n_2(1 - n_2 + a_{21}n_1) = g(n_1, n_2).$$

5.4.2 Steady states

This model has steady-states $(0, 0)$, $(1, 0)$,$(0, 1)$ and

$$({n_1}^*, {n_2}^*) = \left(\frac{1 + a_{12}}{1 - a_{12}a_{21}}, \frac{1 + a_{21}}{1 - a_{12}a_{21}}\right).$$

The coexistence steady state is biological relevant if $a_{12}a_{21} < 1$

5.4.3 Nullclines

The n_1 nullclines satisfy

$$n_1 = 0, \quad n_2 = \frac{1}{a_{12}}(n_1 - 1).$$

The n_2 nullclines satisfy

$$n_2 = 0, \quad n_2 = 1 + a_{21}n_1.$$

Note that both nullclines have a positive slope.

5.4.4 Linear stability

Using a similar analysis to the competition model, it can be shown that the steady states (0,0), (1,0) and (0,1) are unstable. When $a_{12}a_{21} < 1$ there is a stable coexistence steady state. See Figure 5.5.

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



n_1_vec=np.linspace(0,5,100)
```

```
def rhs_sym_model(x,t):
  rhs=np.zeros_like(x)
  n_1=x[0]
  n_2=x[1]
  dn_1_dt=n_1*(1-n_1)+a_12*n_1*n_2
  dn_2_dt=rho*(n_2*(1-n_2)+a_21*n_1*n_2)
  rhs[0]=dn_1_dt
  rhs[1]=dn_2_dt
  return rhs

def ComputeSymbiosisSol(a_12,a_21,n_1_vec):
    t = np.linspace(0, 6, 1000)

    init_cond1=[0.75,0.75]
    init_cond2=[0.15,0.15]
    init_cond3=[2.5,0.5]

    alpha=2.0
    sol1 = odeint(rhs_sym_model, init_cond1,t)
    sol2 = odeint(rhs_sym_model, init_cond2,t)
    sol3 = odeint(rhs_sym_model, init_cond3,t)

    num_steady_states=3
    fourth_ss_condition= ((a_12*a_21<1))
    if fourth_ss_condition==True:
        num_steady_states=4

    ss=np.zeros((num_steady_states,2),dtype=float)
    ss[0,:]=[0,0]
    ss[1,:]=[1,0]
    ss[2,:]=[0,1]
    if fourth_ss_condition==True:
        ss[3,:]=[(1+a_12)/(1-a_12*a_21),(1+a_21)/(1-a_12*a_21)]


    n1_ncline_1_n_1=[0,0]
    n1_ncline_1_n_2=[0,5]
    n1_ncline_2_n_2=1/a_12*(n_1_vec-1)
    n2_ncline_1_n_1=[0,5]
    n2_ncline_1_n_2=[0,0]
    n2_ncline_2_n_2=1+a_21*(n_1_vec)
```

```
    return sol1,sol2,sol3,ss,n1_ncline_1_n_1,n1_ncline_1_n_2,n1_ncline_2_n_2,n2_ncline_1_n


a_12=0.5
a_21=0.4
sol1,sol2,sol3,ss, n1_ncline_1_n_1,n1_ncline_1_n_2,n1_ncline_2_n_2,n2_ncline_1_n_1,n2_ncli

fig, ax = plt.subplots(1,2)
ax[0].plot(n1_ncline_1_n_1,n1_ncline_1_n_2,'k--')
ax[0].plot(n_1_vec,n1_ncline_2_n_2,'k--')
ax[0].plot(n2_ncline_1_n_1,n2_ncline_1_n_2,'r--')
ax[0].plot(n_1_vec,n2_ncline_2_n_2,'r--')

ax[0].plot(sol1[:,0],sol1[:,1],sol2[:,0],sol2[:,1],sol3[:,0],sol3[:,1])
ax[0].plot(ss[:,0],ss[:,1],'k*')
plt.xlabel('$n_1$')
plt.ylabel('$n_2$')

ax[0].set_xlim([-0.05,2.5])
ax[0].set_ylim([-0.05,2.5])

a_12=1.5
a_21=0.8
sol1,sol2,sol3,ss, n1_ncline_1_n_1,n1_ncline_1_n_2,n1_ncline_2_n_2,n2_ncline_1_n_1,n2_ncli
ax[1].plot(n1_ncline_1_n_1,n1_ncline_1_n_2,'k--')
ax[1].plot(n_1_vec,n1_ncline_2_n_2,'k--')
ax[1].plot(n2_ncline_1_n_1,n2_ncline_1_n_2,'r--')
ax[1].plot(n_1_vec,n2_ncline_2_n_2,'r--')

ax[1].plot(sol1[:,0],sol1[:,1],sol2[:,0],sol2[:,1],sol3[:,0],sol3[:,1])
ax[1].plot(ss[:,0],ss[:,1],'k*')
plt.xlabel('$n_1$')
plt.ylabel('$n_2$')

ax[1].set_xlim([-0.05,2.5])
ax[1].set_ylim([-0.05,2.5])


plt.grid()
plt.show()
```

```
/Users/pmurray/anaconda3/lib/python3.11/site-packages/scipy/integrate/_odepack_py.py:248: ODI

Excess work done on this call (perhaps wrong Dfun type). Run with full_output = 1 to get quar
```

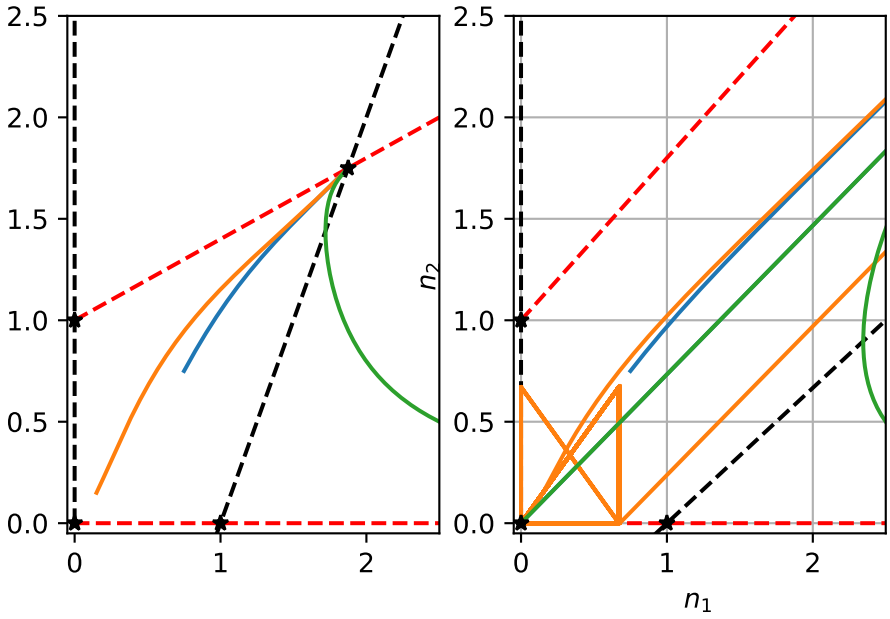

Figure 5.5: Numerical solution of the symbiosis model

5.4.5 Insight

The important parameter in the model is the product $a_{12}a_{21}$. This quantifies the total amount of cooperativity in the model. All steady states that involve the extinction of a species are unstable. In the case where if $a_{12}a_{21} < 1$ there is a stable steady state. Note that steady states of both variables are higher than they would be in the absence of the other species. In the $a_{12}a_{21} > 1$ there is no coexistence steady state and both populations grow in an unbounded manner.