4 Modelling the spread of an epidemic: SIR models

The SIR model describes the spread of an epidemic through a large population. It does this by describing the movement of the population through three phases of the disease: those individuals who are *susceptible*, those who are *infectious*, and those who have been *removed* from the disease. Those individuals in the removed class have either died, or have recovered from the disease and are now immune to it. If the outbreak occurs over a short period of time, we may reasonably assume that the total population is fixed, so that $S'(t) + I'(t) + R'(t) = 0$. We may also assume that $S(t) + I(t) + R(t) = 1$, so that $S(t)$ represents the *fraction* of the population that is susceptible, etc.

Individuals may move from one class to another as described by the flow

$$S \to I \to R.$$

Let us consider the transition rate between S and I. Let β represent the average number of contacts made per day that could spread the disease. The proportion of these contacts that are with a susceptible individual is $S(t)$. Thus, one infectious individual will on average infect $\beta S(t)$ others per day. Let N represent the total population size. Then we obtain the differential equation

$$\frac{d}{dt}(S(t)N) = -\beta S(t)(I(t)N)$$

Now consider the transition rate between I and R. We assume that there is a fixed proportion γ of the infectious group who will recover on a given day, so that

$$\frac{d}{dt}R(t) = -\gamma I(t).$$

Note that γ is the reciprocal of the average length of time spent in the infectious phase.

Since the derivatives sum to 0, we have $I'(t) = -S'(t) - R'(t)$, so the differential equations are given by

$$\frac{dS}{dt} = -\beta IS,$$
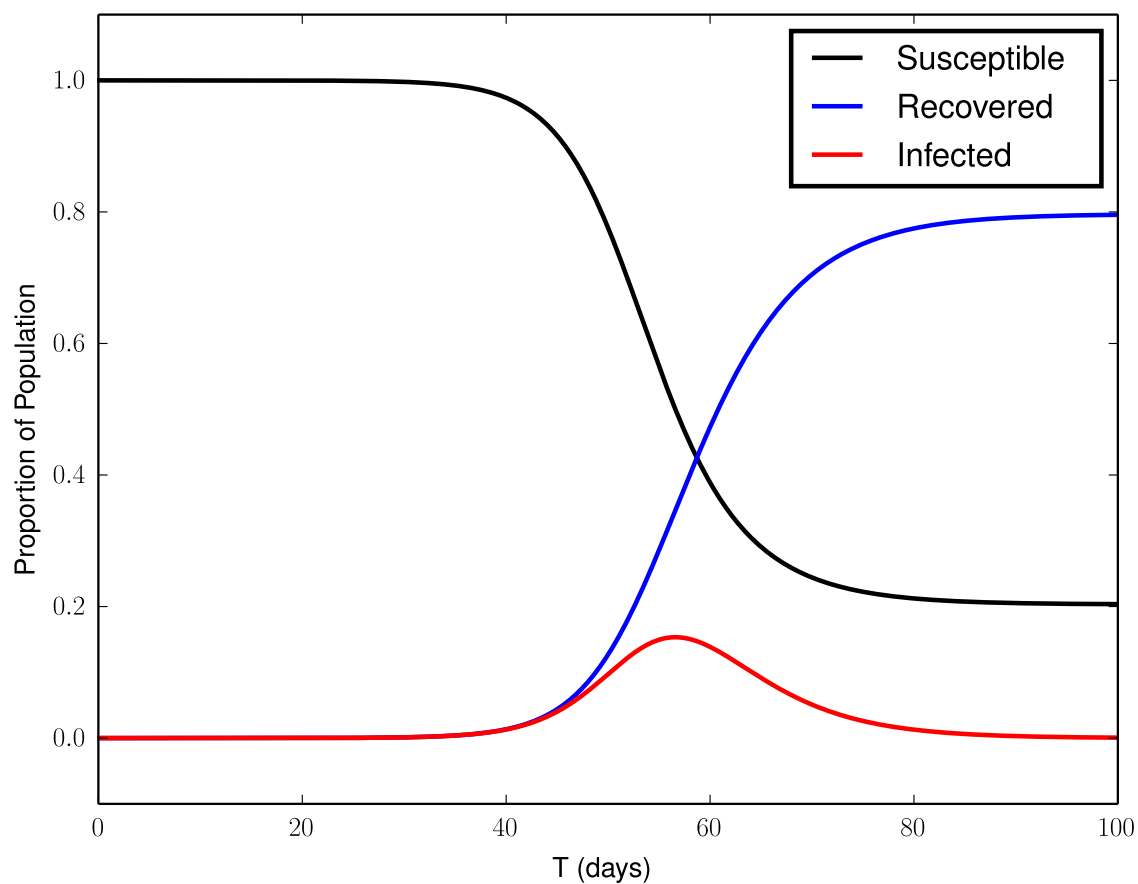$$\frac{dI}{dt} = \beta IS - \gamma I,$$
$$\frac{dR}{dt} = \gamma I.$$

Figure 4.1: Solution to Problem (1)

Problem 1. Solve the IVP

$$\frac{dS}{dt} = -\frac{1}{2}IS,$$

$$\frac{dI}{dt} = \frac{1}{2}IS - \frac{1}{4}I,$$

$$\frac{dR}{dt} = \frac{1}{4}I,$$

$$S(0) = 1 - 6.25 \cdot 10^{-7},$$

$$I(0) = 6.25 \cdot 10^{-7},$$

$$R(0) = 0,$$

on the interval $[0, 100]$, and plot your results. See Figure 4.1.

Problem 2. Suppose that, in a city of approximately three million, five have recently entered the city carrying a certain disease. (Suppose they have just entered the infectious state.)

Each of those individuals has a contact each day that could spread the disease, and an average of three days is spent in the infectious state. Find the solution of the corresponding SIR equations for the next fifty days and plot your results.

At the peak of the infection, how many in the city will still be able to work (Assume for simplicity that those who are in the infectious state either cannot go to work or are unproductive, etc.) Print your result.

Problem 3. Suppose that, in a city of approximately three million, five have recently entered the city carrying a certain disease. (Suppose they have just entered the infectious state.)

Each of those individuals will make three contacts every ten days that could spread the disease, and an average of four days is spent in the infectious state. Find the solution of the corresponding SIR equations and plot your results. See Figure 4.2.
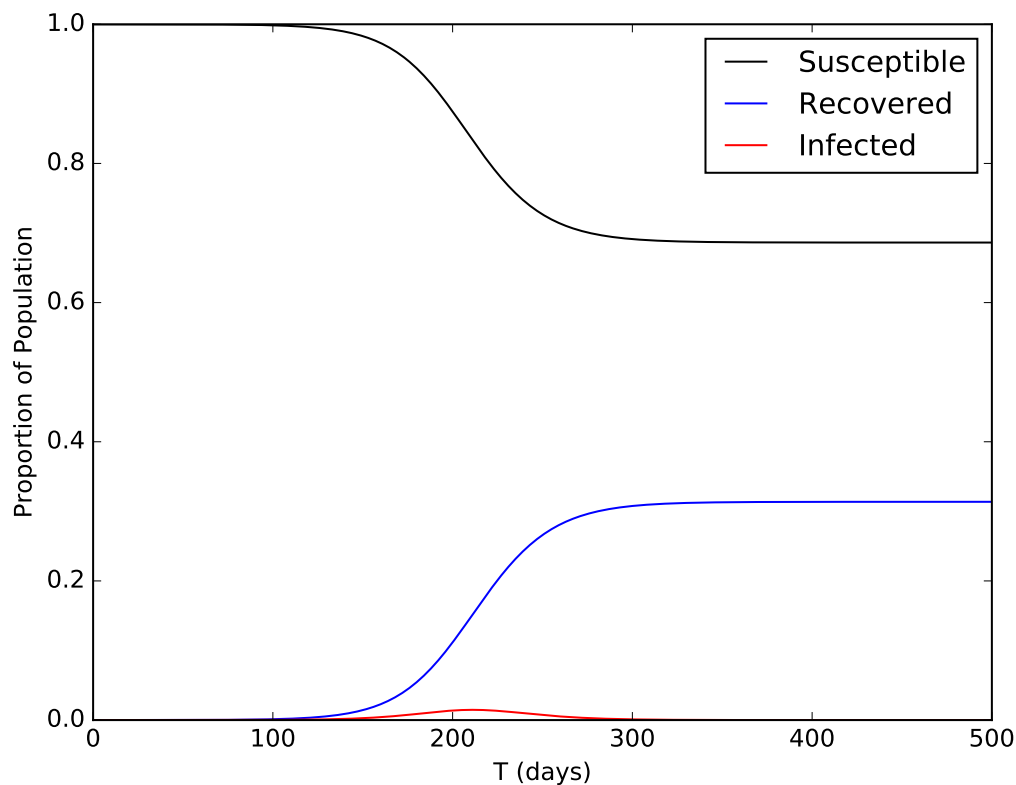

Figure 4.2: Solution to Problem (3).

Variations on the SIR Model

SIS Models describe diseases where individuals who have recovered from the disease do not gain any lasting immunity. There are only two compartments in this model: those who are *susceptible*, and those who are *infectious*.

The basic equations are given by

$$\frac{dS}{dt} = -\beta IS + \gamma I,$$
$$\frac{dI}{dt} = \beta IS - \gamma I$$

If we add to our basic SIR model to account for the death rate and an equal birth rate, the equations become

$$\frac{dS}{dt} = \mu(1 - S) - \beta IS,$$
$$\frac{dI}{dt} = \beta IS - (\gamma + \mu)I,$$
$$\frac{dR}{dt} = \gamma I - \mu R$$

where μ represents the death rate and equal birth rate, noting that any new person born is born into the susceptible population.

SIRS models take the previous model and allow the transfer of individuals from the recovered/removed class to rejoin the susceptible class.

$$\frac{dS}{dt} = fR + \mu(1 - S) - \beta IS,$$
$$\frac{dI}{dt} = \beta IS - (\gamma + \mu)I,$$
$$\frac{dR}{dt} = -fR + \gamma I - \mu R.$$

Problem 4. In the world there are 7 billion people. Influenza, or the flu, is one of those viruses that everyone can be susceptible to, even after recovering from their last sickness. The flu virus is able to change in order to evade our immune system and we become susceptible once more (although technically it is now a different strain). Suppose the virus originates with 1000 people in Texas after Hurricane Harvey flooded Houston and stagnant water allowed the virus to proliferate. According to WebMD (trustworthy source, right?), once you get the virus you are contagious up to a week, and kids up to 2 weeks. For this lab, suppose you are contagious for 10 days before recovering. Also suppose that on average someone makes one contact every two days that could spread the flu. Since we can catch a new strain of the flu, suppose that a recovered individual becomes susceptible again with probability $f = 1/50$. The flu is also known to be deadly, killing hundreds of thousands every year on top of the normal death rate. To assure a steady population, let the birth rate balance out the death rate, and in particular let $\mu = .0001$.

Using the SIRS model above, plot the proportion of population that is Susceptible, Infected, and Recovered over a year span (365 days).

Boundary Value Problem

The next exercise uses a variation of the SIR model called an SEIR model to describe the spread of measles (see [1]). This new model adds another compartment, called the *exposed* or *latency* phase. It assumes that the rate at which measles is contracted depends on the season, i.e. the rate is periodic. That allows us to formulate the yearly occurrence rate for measles as a boundary value problem. The boundary value problem looks like

$$\begin{bmatrix} S \\ E \\ I \end{bmatrix}' = \begin{bmatrix} \mu - \beta(t)SI \\ \beta(t)SI - E/\lambda \\ E/\lambda - I/\eta \end{bmatrix}, \tag{4.1}$$

$$S(0) = S(1), \tag{4.2}$$
$$E(0) = E(1), \tag{4.3}$$
$$I(0) = I(1) \tag{4.4}$$

Parameters μ and λ represent the birth rate of the population and the latency period of measles, respectively. η represents the infectious period before an individual moves from the infectious class to the recovered class. After recovery an individual remains immune, which is why $R(t)$ is not included in the system. The set up of this problem is not normal since we are excluding $R(t)$, but it results in a nice graph.

To solve this problem we will use a full-featured BVP solver that is available in `scipy`. The code below demonstrates how to use `solve_bvp` to solve the BVP

$$\epsilon y'' + yy' - y = 0, \quad y(-1) = 1, \quad y(1) = -1/3, \epsilon = .1 \tag{4.5}$$

Look at figure 4.3 for the solution.

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

epsilon, lbc, rbc = .1, 1, - 1/3

# The ode function takes the independent variable first
# It has return shape (n,)
def ode(x , y):
    return np.array([y[1] , (1/epsilon) * (y[0] - y[0] * y[1])])

# The BVP solver expects you to pass it the boundary
# conditions as a callable function that computes the difference
# between a guess at the boundary conditions
# and the desired boundary conditions.
# When we use the BVP solver, we will tell it how many constraints
# there should be on each side of the domain so that it knows
# how many entries to expect in the tuples BCa and BCb.
```

---

[1] Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, by Aescher, Mattheij, and Russell
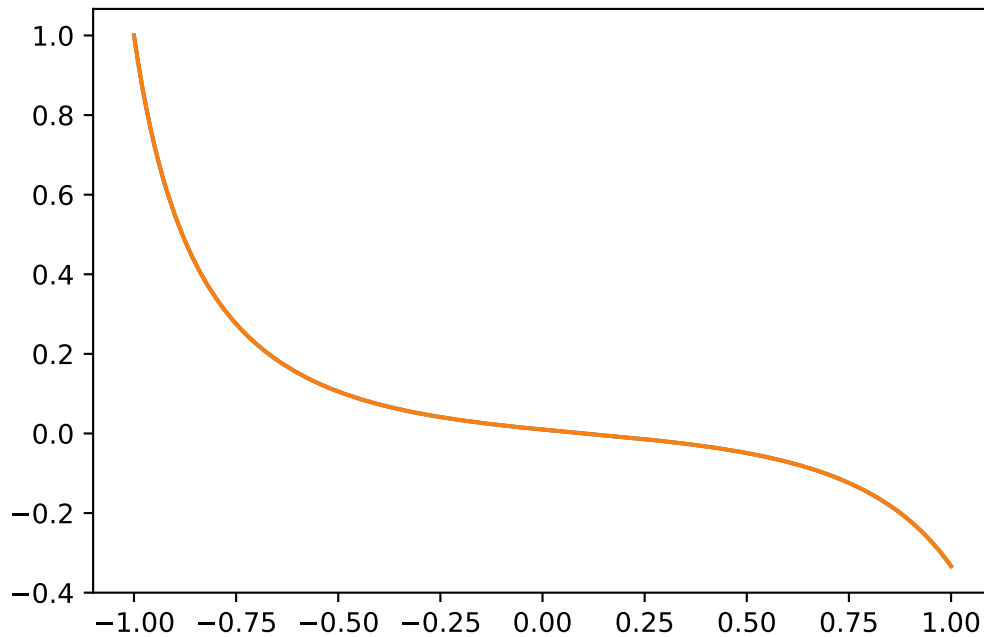
Figure 4.3: Solution to (4.5)

```python
# In this case, we have one boundary condition on either side.
# These constraints are expected to evaluate to 0 when the
# boundary condition is satisfied.

# The return shape of bcs() is (n,)
def bcs(ya, yb):
    BCa = np.array([ya[0] - lbc])    # 1 Boundary condition on the left
    BCb = np.array([yb[0] - rbc])    # 1 Boundary condition on the right
    # The return values will be 0s when the boundary conditions are met exactly
    return np.hstack([BCa, BCb])

# The independent variable has size (m,) and goes from a to b with some step ↵
    size
X = np.linspace(-1, 1, 200)
# The y input must have shape (n,m) and includes our initial guess for the ↵
    boundaries
y = np.array([-1/3, -4/3]).reshape((-1,1))*np.ones((2, len(X)))

# There are multiple returns from solve_bvp(). We are interested in the y ↵
    values which can be found in the sol field.
solution = solve_bvp(ode, bcs, X, y)
# We are interested in only y, not y', which is found in the first row of sol.
```

```
y_plot = solution.sol(X)[0]

plt.plot(X, y_plot)
plt.show()
```

Problem 5. Consider (4.1) Let the periodic function for our measles case be $\beta(t) = \beta_0(1 + \beta_1 \cos 2\pi t)$. Use parameters $\beta_1 = 1$, $\beta_0 = 1575$, $\eta = 0.01$, $\lambda = .0279$, and $\mu = .02$. Note: in this case, time is measured in years, so run the solution over the interval $[0, 1]$ to show a one-year cycle. The boundary conditions are really just saying that the year will begin and end in the same state.

solve_bvp requires *separated boundary conditions*. In other words, each equation in the set of boundary conditions can only include values at one end of the interval. To deal with this, let $C = [C_1, C_2, C_3]$, and add the equation

$$C' = 0$$

to the system of ODEs given above (for a total of 6 equations). Then the boundary conditions can be separated using the following trick:

$$\begin{pmatrix} C_1(0) \\ C_2(0) \\ C_3(0) \end{pmatrix} = \begin{pmatrix} S(0) \\ E(0) \\ I(0) \end{pmatrix}, \quad \begin{pmatrix} C_1(1) \\ C_2(1) \\ C_3(1) \end{pmatrix} = \begin{pmatrix} S(1) \\ E(1) \\ I(1) \end{pmatrix}.$$

Now C_1, C_2, C_3 become the 4th, 5th, and 6th rows of your solution matrix, so the 3 boundary conditions for the left are obtained by subtracting the last three entries of $y(0)$ from the first three entries, giving you $ya[0:3] - ya[3:]$. Similarly, your right boundary conditions will look like $yb[0:3] - yb[3:]$.

When you code your boundary conditions, note that solve_bvp changes the initial conditions to force all the entries in the return of bcs() to be zero. You can use the initial conditions from Fig. 4.4 as your initial guess (which will be an array of 6 elements). Remember that the initial infected proportion is small, not 0.
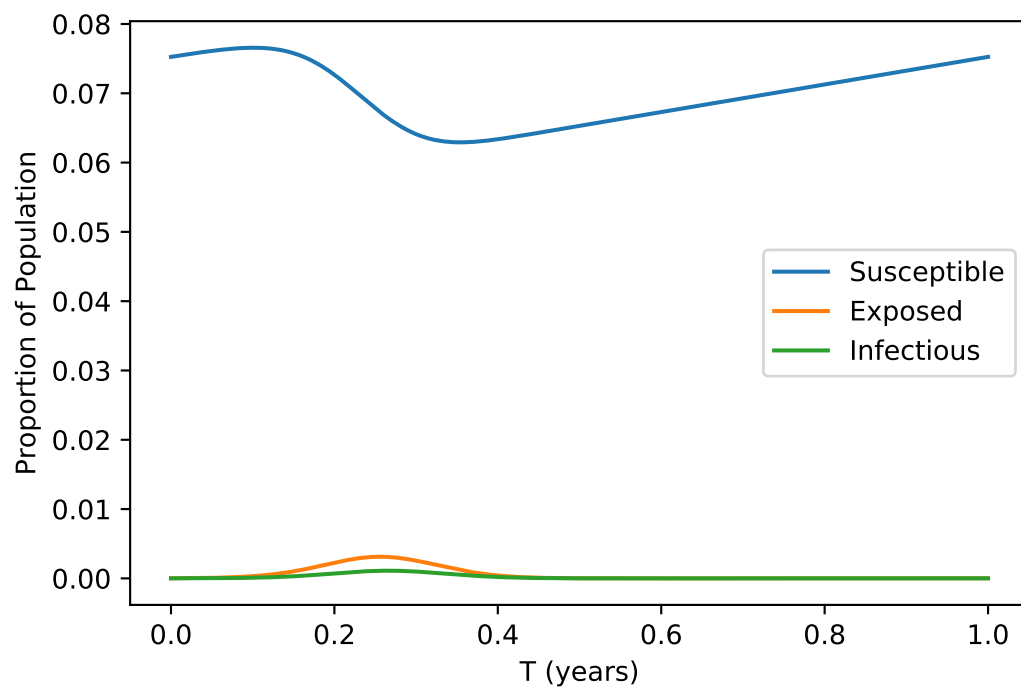
Figure 4.4: Solution to Problem (5)