Peruvian Competence Center for Scientific Computing (PeC^3) www.pec3.org
Dr. S. Frei, Dr. V. Knittel, P. Spitz (Universität Konstanz)

Programming Project

Solving Models of Mathematical Epidemiology in Python

1 Introduction

This project is a follow-up on the mini-course *Programming with Python - an introduction* held by Dr. Vanessa Knittel at CIMAC-X 2021. It is intended (mainly) for Bachelor and Master students as an introduction to solve mathematical problems (in particular Ordinary Differential Equations) in Python. The objectives are to get to know each other and to give you some insights in the role of programming within Applied Mathematics. Of course, PhD students or lecturers with an interest in the subject are also invited to participate. Note, however, that we will not be able to issue any certificate, as the project work is mainly self-independent.

If you would like to participate, please send a mail to stefan.frei@uni-konstanz.de before **September 10, 2021**, indicating your name, university and current status (for example "8th semester Bachelor student in Mathematics"). We will organize a preliminary (virtual) meeting at beginning of October and be available for support until February, 2022, where a final meeting is planned.

The project has been part of the lecture *Computer usage in Mathematics*, which is part of the first- or second-year Bachelor of Mathematics curriculum at the University Konstanz. Feel free to adjust the exercises given at the end of this sheet to your interest, for example by adjusting parameters to the situation in Peru, etc.

2 Models of Mathematical Epidemiology

In this section we will introduce the models. We will start with the SIR model which is the easiest model available to describe the spread of infectious diseases. We will also shortly discuss the behaviour of the SIR model. Afterwards we will expand our model to the SEIR model which acknowledges that SARS-COV-2 has some incubation period. The SEIR model will developed further into the vSEIR model by including the existence of vaccines. Here we can look at the effect of different vaccination programmes.

2.1 The SIR model

The SIR model splits a total population of N individuals for every time stamp $t \geq 0$ into three disjoint groups:

- The **susceptible** group $S(t)$
- The **infected** group $I(t)$
- The **recovered** group $R(t)$

We assume that these groups partition the total population, i.e. $S(t) + I(t) + R(t) = N$ for every $t \geq 0$.
Next, we will discuss the way these groups interacts with each other and allow for some flow from one group into another.

2.1.1 $S \to I$

First we will consider the passage from the susceptible group into the infected group. Let $\beta > 0$ be the number of susceptible persons an infected person can infect in a single unit of time given all contacts are with susceptible individuals. Obviously β depends on the infectiousness of the disease and the number of contacts a person has on average.
As not all contacts will be susceptible due to vaccination or prior infection one has to introduce the corrective factor $\frac{S(t)}{N}$. Therefore a single infected individual will infect $\beta\frac{S(t)}{N}$ additional persons per unit of time so the

total flow from S to I is given by

$$\begin{aligned} S'(t) &= -\beta \frac{S(t)}{N} I(t) \\ I'(t) &= \beta \frac{S(t)}{N} I(t) + \dots \end{aligned}$$

In the next section we will discuss the flow from I to R so the rate of change of I is not completely known yet.

2.1.2 $I \to R$

Infected persons will not stay infected forever and will transition into the recovered group. For the SIR model we assume that every unit of time a proportion $\gamma \in (0,1]$ of the infected individuals recover. This yields

$$R'(t) = \gamma I(t)$$

For the change of $I(t)$ we arrive at

$$I'(t) = \beta \frac{S(t)}{N} I(t) - \gamma I(t)$$

γ coheres with the average duration of an infection t_{inf}:

$$\gamma = t_{\text{inf}}^{-1}$$

2.1.3 Assembly of the model

We obtain a system of ordinary differential equations:

$$\begin{aligned} S'(t) &= -\beta \frac{S(t)}{N} I(t) \qquad (1) \\ I'(t) &= \beta \frac{S(t)}{N} I(t) - \gamma I(t) \qquad (2) \\ R'(t) &= \gamma I(t) \end{aligned}$$

This model implies $\partial_t \left(S(t) + I(t) + R(t)\right) = 0$ so the total population stays constant. Therefore $R(t) = N - S(t) - I(t)$ holds and one can omit R in this system of ODEs.
Furthermore one can discuss whether β and γ change with time. As γ corresponds to the average duration of the disease and is therefore only dependent on the genetic make-up of the virus one can assume that γ is constant.
As β depends on the infectiousness and on the number of average contacts this parameter will change over time, e.g. due to lockdowns. In order to model measures to reduce contact we will split $\beta(t)$ into two parameters $\beta(t) = \beta_0 \left(1 - k(t)\right)$ where β_0 corresponds to the infectiousness of the disease without any counter measures and $k(t)$ describes the reduction of contacts.

2.1.4 Normalization

In order to simplify the model we will normalize S and I:

$$\begin{aligned} s(t) &:= \frac{S(t)}{N} \\ i(t) &:= \frac{I(t)}{N} \end{aligned}$$

By combining these definitions with (1) and (2) we obtain

$$\begin{aligned} s'(t) &= -\beta s(t) i(t) \qquad (3) \\ i'(t) &= \beta s(t) i(t) - \gamma i(t) \qquad (4) \end{aligned}$$

In order to ensure a unique solution we have to give initial values $s_0, i_0 \in [0,1]$ with $s_0 + i_0 = 1$.

2.1.5 Behaviour of the model

Finally we will outline the behaviour of the system of ODEs given in (3) and (4).

Theorem 2.1
Let (s, i) be a solution to (3) and (4) with initial values i_0, s_0. Let $\sigma_0 := \beta t_{inf}$.
If $\sigma_0 s_0 \leq 1$ holds then i is monotonically decreasing and $\lim_{t\to\infty} i(t) = 0$.
If $\sigma_0 s_0 > 1$ then i is monotonically increasing until i reaches the value

$$i_{max} = i_0 + s_0 - \frac{1 + \ln(\sigma_0 s_0)}{\sigma_0}$$

After reaching this threshold i will be decreasing monotonically.

Proof. [3, Theorem 2.2] □

2.2 The SEIR model

The SIR model assumes that a newly infected person is instantly contagious which is not true. In order to fix this we introduce a new class, the **exposed** group E where members are freshly infected but not yet contagious individuals. This class will be an intermediate step when transitioning from S to I.

2.2.1 The adjusted flows

For the number of newly infected people the term $\beta\frac{S(t)}{N}I(t)$ still seems reasonable but now the newly infected individuals are placed in E. This yields

$$\begin{aligned} S'(t) &= -\beta\frac{S(t)}{N}I(t) \\ E'(t) &= \beta\frac{S(t)}{N}I(t) - \dots \end{aligned}$$

We introduce another parameter t_{exp} that describes the average latent period. The transitions $E \to I$ and $I \to R$ are similar. Setting $\kappa := \frac{1}{t_{\text{exp}}}$ we have

$$\begin{aligned} E'(t) &= \beta\frac{S(t)}{N}I(t) - \kappa E(t) \\ I'(t) &= \kappa E(t) - \gamma I(t) \\ R'(t) &= \gamma I(t) \end{aligned}$$

Again we can deduce that $S(t)+E(t)+I(t)+R(t)$ is constant and therefore $R(t)$ is omittable. By normalization $e(t) := \frac{E(t)}{N}$ we arrive at

$$\begin{aligned} s'(t) &= -\beta s(t)i(t) \\ e'(t) &= \beta s(t)i(t) - \kappa e(t) \\ i'(t) &= \kappa e(t) - \gamma i(t) \end{aligned}$$

2.3 The vSEIR model

Now we want to include vaccines in our model. We assume that a vaccination instantly transports an individual from the susceptible into the recovered group. We also assume that only susceptible individuals receive vaccinations and that every available vaccine is consumed immediately.
We denote the number of vaccines that can be given out per day with $v(t)$. The number of newly vaccinated persons is given by $\min(S(t), v(t))$ and therefore the system of ODEs can be augmented into

$$\begin{aligned} S'(t) &= -\beta\frac{S(t)}{N}I(t) - \min(S(t), v(t)) \\ E'(t) &= \beta\frac{S(t)}{N}I(t) - \kappa E(t) \\ I'(t) &= \kappa E(t) - \gamma I(t) \\ R'(t) &= \gamma I(t) + \min(S(t), v(t)) \end{aligned}$$

We can normalize this model as before by normalizing the number of available vaccines with $v_{\text{rel}}(t) := \frac{v(t)}{N}$. Again R can be omitted as before. This results in

$$\begin{aligned} s'(t) &= -\beta s(t)i(t) - \min(s(t), v_{\text{rel}}(t)) \\ e'(t) &= \beta s(t)i(t) - \kappa e(t) \\ i'(t) &= \kappa e(t) - \gamma i(t) \end{aligned}$$

For further reference, we refer to the literature, for example [1, 4]

3 Numerical strategies

As already the SIR model is a nonlinear system of ODEs it is in general not possible to find a solution analytically. Here we will develop a simple method to approximate the solution numerically.

3.1 Forward Euler method

In order to illustrate the general method we will start with the simpliest model, the SIR model. We consider an interval $[0, T]$ and start by discretising it into $n + 1$ points in time. For the sake of simplicity we choose an equidistant discretisation which yields the points in time

$$t_k := \frac{T}{n}k, \quad k = 0, 1, \dots, n$$

As we need some approximation for $s'(t_k)$ we pick

$$s'(t_{k-1}) \approx \frac{s(t_k) - s(t_{k-1})}{t_k - t_{k-1}} \tag{5}$$

By using a Taylor series at $t = t_{k-1}$ one can deduce

$$s'(t_{k-1}) - \frac{s(t_k) - s(t_{k-1})}{t_k - t_{k-1}} = -\frac{T}{2n}s''(\xi_k)$$

with $\xi_k \in [t_{k-1}, t_k]$. One can easily show that $s \in C^\infty([0, T])$ and therefore s'' is bounded. Therefore the approximation proposed in (5) has order $\mathcal{O}(n^{-1})$.
Set $s_k := s(t_k)$ and $i_k := i(t_k)$ for every $k = 0, 1, \dots, n$. One can solve (5) for $s(t_k) = s_k$ and yields

$$s_k \approx s_{k-1} + (t_k - t_{k-1})s'(t_{k-1})$$

As (s, i) solves the ODE one has $s'(t_{k-1}) = -\beta s_{k-1}i_{k-1} =: f_s(t_{k-1}, s_{k-1}, i_{k-1})$ and $i'(t_{k-1}) = \beta s_{k-1}i_{k-1} - \gamma i_{k-1} =: f_i(t_{k-1}, s_{k-1}, i_{k-1})$ where the right sides of the ODEs have been hidden in the functions f_s and f_i. By combining this with the approximaton (5) and by using $t_k - t_{k-1} = \frac{T}{n}$ we get

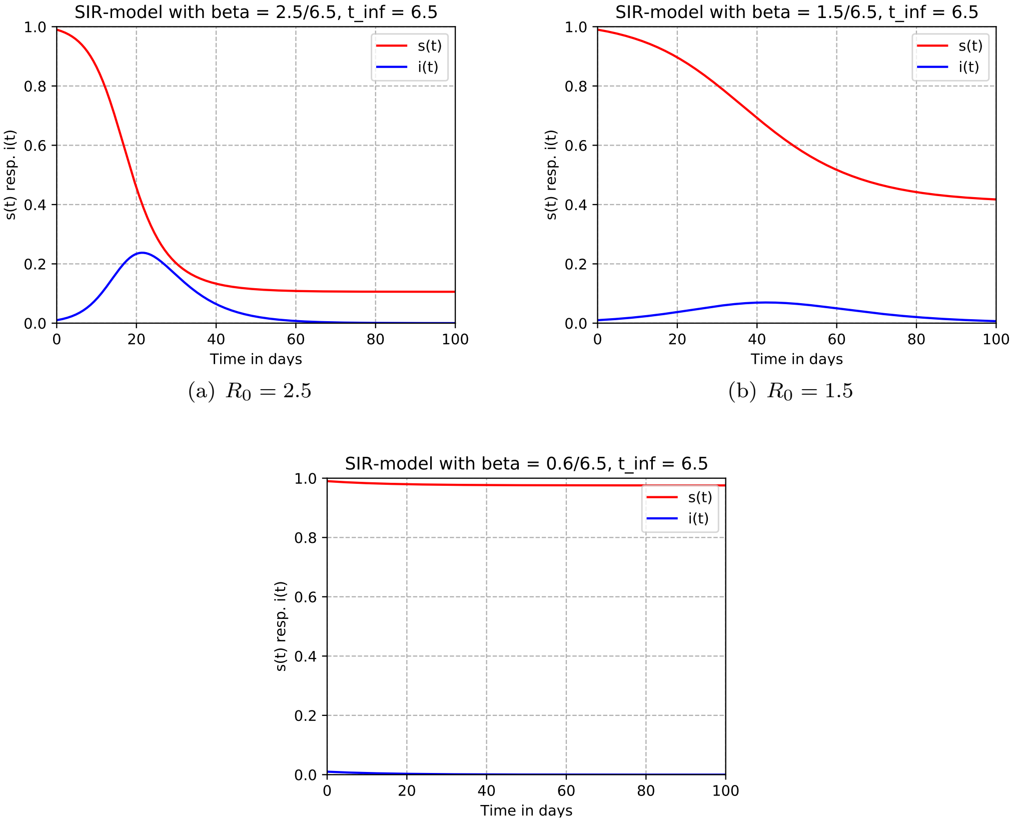
$$\begin{aligned} s_k &\approx s_{k-1} + \frac{T}{n}f_s(t_{k-1}, s_{k-1}, i_{k-1}) \\ i_k &\approx i_{k-1} + \frac{T}{n}f_i(t_{k-1}, s_{k-1}, i_{k-1}) \end{aligned}$$

This recursive formula can easily be implemented in a programming language of your choice, an algorithm is given in algorithm 1.

Algorithm 1: Forward Euler for SIR

```
Input  : Initial values $s_0, i_0 \in [0,1]$, parameters $\beta, \gamma \in \mathbb{R}_+$, grid size $n \in \mathbb{N}$, final time $T$
Output: Approximations $s_k, i_k$ to $s(t_k), i(t_k)$ for $k = 1, \dots, n$
Set k = 1; while $k \leq n$ do
    $s_k = s_{k-1} + \frac{T}{n}f_s(t_{k-1}, s_{k-1}, i_{k-1})$;
    $i_k = i_{k-1} + \frac{T}{n}f_i(t_{k-1}, s_{k-1}, i_{k-1})$;
    $k = k + 1$;
```

The result of this algorithm for different sets of parameters is shown in fig. 1.



(a) $R_0 = 2.5$

(b) $R_0 = 1.5$

(c) $R_0 = 0.6$

Figure 1: Behaviour of the SIR model for different parameters, $i_0 = 0.01$.

3.1.1 Forward Euler method in a general setting

Now we want to generalise algorithm 1 to an arbitrary system of non-autonomous first-order ordinary differential equations, that is

$$x'(t) = F(t, x(t)) \tag{6}$$

$$x(0) = x_0 \tag{7}$$

where x is $\mathbb{R}^n$-valued. As before we discretise the interval $[0, T]$ into $n + 1$ points of time, for sake of simplicity we pick equidistant points. The resulting points in time are again

$$t_k := k\frac{T}{n} \qquad \text{for } k = 0, 1, \ldots, n$$

Denote $x_k := x(t_k)$. With the same argument as before we obtain

$$\begin{aligned} F(t_{k-1}, x_{k-1}) &= x'(t_{k-1}) \\ &\approx \frac{x_k - x_{k-1}}{t_k - t_{k-1}} \end{aligned}$$

Again by solving for x_k one arrives at the recursive formula

$$x_k = x_{k-1} + (t_k - t_{k-1})F(t_{k-1}, x_{k-1})$$

Now one can adjust Algorithm 1 accordingly:

Algorithm 2: General forward Euler

```
Input  : Initial value $x_0 \in \mathbb{R}^n$, right side $F : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$, grid size $n \in \mathbb{N}$, final time $T$
Output: Approximations $x_k$ to $x(t_k)$ for $k = 1, \ldots, n$
Set k = 1; while $k \leq n$ do
    $x_k = x_{k-1} + \frac{T}{n}F(t_{k-1}, x_{k-1})$
    $k = k + 1$;
```

3.2 Total number of infections

In Exercise 6 the total number of infections N_{inf} up to some time τ will be computed. For the SIR- and SEIR model this is an easy task as an individual did not suffer from the disease if and only if it is in group S. For the vSEIR model this turns out to be different as vaccinated individuals are part of group R but weren't infected so the usual formula $N_{\text{inf}} = N - S(\tau)$ does not work. We propose this algorithm to determine N_{inf}:

```
Algorithm 3: Total number of infections
  Input  : Solution $y = (s, e, i) \in \mathbb{R}^{3\times(n+1)}$ taken from forward Euler with final time $T$, vSEIR-parameter
           $\kappa$, total population $N$, time of evaluation $\tau$
  Output: Total number of infections $N_{\text{inf}}$ up to time $\tau$
  Set $i_{\text{total}} := i_1$, $h := \frac{T}{n}$, $k := 1$;
  while $hk < \tau$ do
      $i_{\text{total}} = i_{\text{total}} + h\kappa e_k$
      $k = k + 1$;
  Set $N_{\text{inf}} := N i_{\text{total}}$
```

4 Exercises

These exercises aim to solve the vSEIR-model using the forward Euler method. First you are going to implement a general forward Euler routine, test it with some example scenarios and "experimentally" confirm Theorem 2.1. Then you will create some routines to load parameters without changing your source code. After that you will start exploring the vSEIR-model in detail, i.e. you will calculate the number of total infections, analyze the progress of the pandemic using phase portraits and consider the impact of different vaccination programmes. Finally you will evaluate a simple lockdown strategy using a modified Euler algorithm.

1. Implement algorithm 2 in a python function `y = forward_euler(fun, y_0, T, n)`. Here fun is supposed to be a function representing the right-hand side of the ODE akin to F in (6). y_0 denotes the initial value, T denotes the final time and n describes the grid size.

2. Use the function `forward_euler` to solve the SIR-model with parameters $\beta_0 = 0.5$, $\gamma = 0.15$ and $k \equiv 0.3$ respectively $k \equiv 0.6$. Pick a sufficient final time T and grid size n. Try different initial values $(s_0, i_0) \in [0,1]^2$ with $s_0 + i_0 = 1$.

3. Pick four scenarios of your choice. For each scenario plot s and i in a single graph. Add axis labels and a legend to each graph.
   Compare different initial values and different values of k. Confirm Theorem 2.1.

4. Now write a function that reads the file `Param.txt`. This file contains a set of parameters for the vSEIR-model, taylored to the situation of Germany and taken (mainly) from [2, Table 1]. Every line consists of the name of the parameter, a single space and then the corresponding number. Store the parameters in a dictionary `param` where the keys are the names of the parameters and the values are the corresponding values.
   Read the files `Test1.txt` and `Test2.txt` analogously and store the resulting parameters and initial values in dictionaries `test_1` resp. `test_2`. Print every dictionary to the console.

5. Use `forward_euler` to solve the SIR-model with the settings given by `Test1.txt`. Solve the vSEIR-model using the settings provided by `Test1.txt` and `Test2.txt`. Construct a third setting `test_3` of your choice with non-constant k and v_{rel} and test it as well.
   Plot the resulting curves for i for all test settings in a single graph. Compare the behaviour of the SIR- and the vSEIR-model for setting `test_1`. What is the effect of the class e? In what way do different values of k and v_{rel} change the dynamics of the model?

6. Implement algorithm 3 in a function `N_inf = total_infected(y, kappa, N, T, t)`. Calculate the total number of infected people up to some time τ for setting `test_1` where τ is up to your choice. Compute $i(\tau) + r(\tau)$ in the same setting. Why do those values coincide?
   Perform the same calculations for the settings `test_2` and `test_3`. Do the results of $i(\tau) + r(\tau)$ and `total_infected` coincide? If not, why?

7. Consider now the settings `test_2` and `test_3`. Compute N_{inf} for every single day and save the resulting data to the files `totalInf2.txt` and `totalInf3.txt`.

8. Create phase portraits akin to Fig. 2 for every setting. Choose s as x-axis and i as y-axis. Draw the trajectories for different initial values (s_0, i_0). You may omit the starting values provided by the particular setting and you may use the tool provided in `DrawArrows.py`. Interpret every phase portrait.

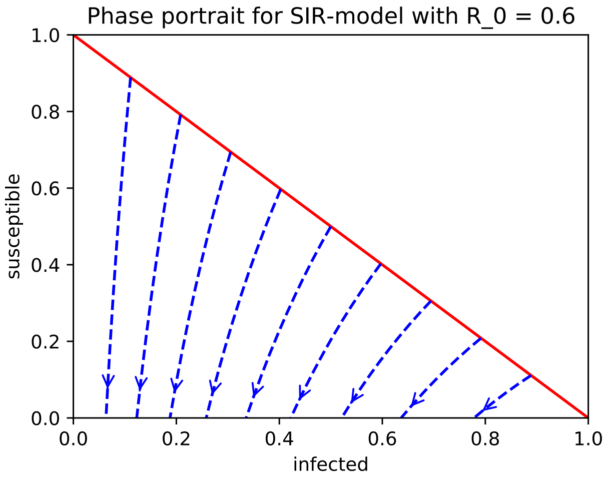


Figure 2: Example phase portrait for SIR

9. For the final exercise we want to analyze the behavior of the vSEIR-model in an adaptive lockdown strategy. We assume a simple lockdown strategy depending on the number of active infections:
   If there is no active lockdown we have $k(t) = k_a$, during an active lockdown we have reduced contacts and $k(t) = k_b$ with $0 \leq k_a < k_b \leq 1$. We start our simulation without an active lockdown. Once the number of infectious individuals reaches or surpasses a threshold i_b we start a lockdown until the number of infectious individuals decreases below i_a where $0 < i_a < i_b \leq 1$.
   Write a pseudocode that is suitable for simulating this strategy involving the vSEIR-model. Implement your pseudocode in a function `adaptive_euler`.
   Pick $k_a = 0.4$, $k_b = 0.8$ and i_a and i_b of your choice. Plot and analyse the resulting curves for every scenario. Do the lockdown strategies prove useful or could you think of a different strategy? How do vaccines influence the situation?

References

[1] Fred Brauer, Carlos Castillo-Chavez, and Zhilan Feng. *Mathematical Models in Epidemiology*, volume 69. Springer, New York, 2019.

[2] Sebastian Contreras, Jonas Dehning, Sebastian B. Mohr, Simon Bauer, F. Paul Spitzner, and Viola Priesemann. Low case numbers enable long-term stable pandemic control without lockdowns, 2021. arXiv e-print 2011.11413, url: `https://arxiv.org/abs/2011.11413`.

[3] Herbert W. Hethcote. Qualitative analyses of communicable disease models. *Mathematical Biosciences*, 28(3):335–356, 1976.

[4] Herbert W. Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.