Parameter Estimation in the SIR Model in Mathematical Epidemiology

Bachelor Thesis

Boan Song

Department of Mathematics and Statistics
University of Konstanz

Supervised by Prof. Dr. Stefan Volkwein

Konstanz, 2022
Acknowledgements

I would like to thank all the people who have helped me with my work. I’m extremely grateful to Dr. Stefan Frei for his great advice and guidance during the entire processing time of this thesis. He is always willing to answer my questions. I am also thankful to Prof. Stefan Volkwein for the opportunity to write my bachelor’s thesis in the field of Numerical Optimization and for advice on my thesis. I’d also like to acknowledge my parents for their support.
1 Introduction

In December 2019, viral pneumonia caused by the coronavirus (COVID-19) emerged and rapidly swept the world. On March 12, 2020, the World Health Organization (WHO) declared that COVID-19 could be described as a pandemic. [7] Nowadays, it is still spreading, with the total number of cases in the world reaching 608 million, and in Germany reaching 32 million according to data released by Johns Hopkins University on September 10, 2022. [5] At the same time, COVID-19 has affected the lives of people around the world and has become a major concern. Many countries have experienced city closures, unemployment, economic shocks, and loss of life. It is well known that mathematical models are an effective way to approach epidemic infectious diseases. Therefore, this thesis will start with the construction of the susceptibility-infection-recovery (SIR) model, which is one of the common compartmental models. This model describes the variation of population compartments [4]: Susceptible people become infected through contact with an infected person. And every infected person goes through the same disease process. Eventually, the person withdraws from the process due to death or recovery. [8] The model can be widely used to measure changes in parameters, analyze the impact of parameters, and predict epidemics. It is valuable for the government to implement prevention and control strategies more effectively. [9] Here, we investigate the current outbreak of coronavirus in Germany with the model. The COVID-19 infectious disease data from 1 January to 27 February 2022 in Germany [5] will be applied to the numerical simulations and the results of the parameter estimation will be analyzed. The process of parameter estimation of contact rate $\beta$, which is one of the most important parameters influencing pandemic transmission, will be performed by least squares optimization. There are many ways to solve the least squares problem, here we use the gradient method which is described in section 3.1. Gradient descent is a commonly used iterative first-order derivative optimization algorithm to find optimal solutions to various problems. We use gradient descent here by iteratively adjusting the parameters (i.e., contact rate $\beta$) to minimize the sum-of-residuals function in the least squares problem. That is, the value of the function is gradually reduced by an appropriate stepsize until converges to the minimum value of the function. To complete the algorithm, the first-order derivative of the given function needs to be calculated. In this process, the sensitivity equation from Section 3.2 will be used. Then we apply the explicit Euler method to calculate all the discretization. The constant contact rate is first estimated, and the piecewise constant contact rate is also estimated in order to have more application scenarios. In addition, we varied the turning points to obtain a better function fit.

2 SIR model

The SIR model is one of the simplest epidemiological models. It divides the affected population at each time point $t$ into 3 states:

$S(t)$ : Susceptible, who has never been infected.
\(I(t)\) : Infectious, who is infected and has the capability of spreading the disease.

\(R(t)\) : Recovered, infected people who get cured or die.

\[N = S(t) + I(t) + R(t).\] (1)

[1] The population is considered constant because the course of the epidemic is short compared to a lifetime. [8] This model will not take into account birth and death rates as they have little effect over a short period. People who are infected will be assumed to have recovered at the end of the disease and have immunity, meaning that people will not be repeatedly infected by the disease. [1]

Figure 1 shows the dynamic process between \(S\), \(I\), and \(R\). A description of the individual numerical dynamics in these three states, i.e. the derived differential equations, will be presented later. The derivation of the model and discretization below is based on [1].

Figure 1: SIR model schematic diagram

2.1 Derivation of the model

To accurately explain the SIR model, two parameters \(\beta\) and \(\gamma\) will be added here. \(\beta\) denotes the contact rate, while \(I(t)/N\) denotes the proportion of infected individuals at time point \(t\). Concerning the transmission from \(S\) to \(I\), the average number of contacts of infected individuals with others at time point \(t\) is \(\beta I(t)/N\), which also indicates the probability of an individual being infected in \(S\). Over the entire population, therefore, the differential equation

\[\dot{S}(t) = -\beta I(t) S(t)\] (2)

means individuals infected, and accordingly obtain \(\dot{I}(t) = \beta I(t) S(t)\). Let \(1/\gamma\) be the average infectious period. With recovery rate \(\gamma\), the transition from \(I\) to \(R\) will be described as

\[\dot{R}(t) = \gamma I(t)\] (3)

Therefore, by subtracting \(\gamma I(t)\) from \(\dot{I}\), it becomes

\[\dot{I}(t) = \beta I(t) S(t) - \gamma I(t).\] (4)

Due to the independence of \(S, I\) and \(R\), the differential equations can be reduced to (2) and (4), after transforming (1) to

\[R(t) = N - S(t) + I(t).\] (5)
By normalizing the variables \( s(t) := \frac{S(t)}{N}, \ i(t) := \frac{I(t)}{N} \) from equations (2) and (4), leads to

\[
\begin{align*}
\dot{s}(t) &= -\beta(t)i(t)s(t), \\
\dot{i}(t) &= \beta(t)i(t)s(t) - \gamma i(t).
\end{align*}
\] (6)

In fact, the contact rate depends on many factors, such as season, temperature, government measures, etc., and it cannot still be constant. For example, when the number of infections from the disease reaches a high level during a certain period, the government will set up measures, and people will meet with others less often and wear masks on more occasions in the hope of reducing the risk. This will result in lower exposure rates. So it can be said more precisely that it is a function of time. In later paragraphs, talking about the piecewise parameter, we will assume that it is a function. When the system of equations has the additional start condition \( s(0) = s_0, \ i(0) = i_0 \) at the time point \( t = 0 \), there is a uniquely determined solution \( s(t) \) and \( i(t) \) for \( t \in [0, T] \) due to the rule of analysis III.

### 2.2 Discretization

A numerical solution to the equations is obtained here using the explicit Euler method. Let \( n + 1 \) time points be distributed equidistantly discrete in time interval \([a, b]\) and only \( n \) points will be solved. These discrete time points are obtained by

\[
t_k = a + \frac{b - a}{n} k, \quad k = 0, 1, 2, \ldots, n.
\]

The approximate solutions of the SIR model are constructed from the susceptibility value at time \( t_k \) (denoted as \( s_k \)) and infection value at time \( t_k \) (denoted as \( i_k \)). The equations

\[
\begin{align*}
\dot{s}(t_k) &\approx \frac{s_{k+1} - s_k}{t_{k+1} - t_k}, \\
\dot{i}(t_k) &\approx \frac{i_{k+1} - i_k}{t_{k+1} - t_k}
\end{align*}
\]

at these time instances are through the recursive step

\[
\begin{align*}
s_{k+1} &= s_k + \frac{b - a}{n} f_s(t_k, i_k, s_k), \\
i_{k+1} &= i_k + \frac{b - a}{n} f_i(t_k, i_k, s_k),
\end{align*}
\]

for which \( f_s \) and \( f_i \) are the functions on the right-hand side of the equation system, namely

\[
\begin{align*}
f_s(t_k, i_k, s_k) &= -\beta i_k s_k, \\
f_i(t_k, i_k, s_k) &= \beta i_k s_k - \gamma i_k.
\end{align*}
\] (7)
3 Parameter estimation

We estimate $\beta$ for given data by the basic mathematical least squares problem. This problem is also known as minimizing the sum of squared errors

$$J(\beta, \gamma) = \min_{\beta, \gamma} \frac{1}{2N} \sum_{k=1}^{N} (i(t_k) - \bar{i}_k)^2,$$  \hspace{1cm} (8)

such that

$$\dot{s}(t) = -\beta(t)i(t)s(t),$$

$$\dot{i}(t) = \beta(t)i(t)s(t) - \gamma i(t),$$

where $\gamma$ is constant, with $\gamma_a \leq \gamma \leq \gamma_b$ and $\beta$ is time-dependent, with $\beta_a \leq \beta \leq \beta_b$. The start condition is $s(0) = s_0$, $i(0) = i_0$. We will first focus on the derivative of $J(\beta, \gamma)$ with respect to $\beta$. When the derivative reaches zero, the result of $J(\beta, \gamma)$ gets minimum. We will use the gradient method to find the minimum of the derivative. The sensitivity equations of (6) will be needed in the process.

3.1 Gradient descent method

To solve the unconstrained optimization problem (8), we use the gradient method. The main idea of the method is to find a direction $d \in \mathbb{R}^n$ at a given point $x \in \mathbb{R}^n$ along which the function value of $J(\beta, \gamma)$ falls. Then walk in the direction $d$ until the value of the function $J(\beta, \gamma)$ decreases sufficiently. [3]

**Definition 3.1.** Let $J : \mathbb{R}^n \rightarrow \mathbb{R}$ and $x \in \mathbb{R}$. A direction $d \in \mathbb{R}^n$ is called descent direction of $J$ at the point $x$, when there is a $\alpha_0 > 0$ with

$$J(x + \alpha d) < J(x) \quad \text{for all } \alpha \in (0, \alpha_0]$$  \hspace{1cm} (9)

Let $J$ be continuously differentiable, and

$$\nabla J(x)^T d < 0,$$  \hspace{1cm} (10)

then $d \in \mathbb{R}^n$ is a descent direction of $J$ in $x$. In order to prove this, we will first define $\varphi(\alpha) = J(x + \alpha d)$. Since $J \in C^1(\mathbb{R}^n, \mathbb{R})$, according to Taylor expansion, we can obtain

$$\varphi(\alpha) = \varphi(0) + \alpha \varphi'(0) + r(\alpha) \quad \text{with } \frac{r(\alpha)}{\alpha} \rightarrow 0 \text{ for } \alpha \downarrow 0,$$  \hspace{1cm} (11)

that means $r(\alpha) = O(\alpha)$. Then we can get $\varphi(0) = J(x)$ and $\varphi'(0) = \nabla J(x)^T d < 0$. From (11) we can get

$$\frac{\varphi(\alpha) - \varphi(0)}{\alpha} = \nabla J(x)^T d + \frac{r(\alpha)}{\alpha}.$$
Thus, there exists a $\alpha_0 > 0$ with
\[ \forall \alpha \in (0, \alpha_0) : \frac{\varphi(\alpha) - \varphi(0)}{\alpha} < 0. \]
This also implies that $J(x + \alpha d) < J(x)$ and then we can conclude that $d$ is a descent direction. \[3\] The vector $d = -\nabla J(x)$ is consistently a descent direction because at this point $\nabla J(x)^T d = -||\nabla J(x)||^2 < 0$, unless $x$ is a stationary point. The vector $-\nabla J(x)$ is even the steepest direction of descent. The normalized gradient of $J$ in $x$ is the minimum of $\min \nabla J(x)^T d$ with $||d||^2 = 1$. If $d = -\nabla J(x)$ is chosen, the descent method is called a gradient method. \[3\]

The gradient method is used in this thesis to search the minimum derivative of $J(\beta, \gamma)$ with respect to $\beta$. The detailed algorithm can be seen in subsection 3.6. Among this method, we also need computation of the derivative and the sensitivity equation of (6).

### 3.2 Sensitivity equation

For solving parameter estimation problems for dynamical systems, we consider integration for ordinary differential equations. Since the gradient methods used for least-squares optimization are first-order based, a first-order derivative of the fitting function with respect to the parameters to be estimated is required for them. It is also necessary to have the derivatives of the differential equations since the fitting criterion depends on their solutions. In order to calculate the derivatives of the solutions of the differential equations with respect to the parameters or initial values on the right-hand side, we can solve the so-called sensitivity problem. The following presentation is based on [2]. We consider the $s$-dimensional system (12) with the parameter $p$ from $\mathbb{R}^n$.

\[\begin{align*}
\dot{y}_1 &= F_1(p, y, t), & y_1(0) &= y_1^0(p), \\
\vdots \\
\dot{y}_s &= F_s(p, y, t), & y_s(0) &= y_s^0(p).
\end{align*}\]

The solution of the system for the time variable $t$, the parameter vector $p \in \mathbb{R}^n$ and the initial value
\[ y_0(p) = (y_1^0(p), \ldots, y_s^0(p))^T \in \mathbb{R}^s \]

is denoted by
\[ y(p, y_0(p), t) = (y_1(p, y_0(p), t), \ldots, y_s(p, y_0(p), t))^T \in \mathbb{R}^s, \]

where the initial value also depends on the parameter vector $p$. Now we calculate the solution of the vector $y(p, y_0(p), t)$ and its Jacobian matrix with respect to $p$. The right-hand side $F(p, y, t) = (F_1(p, y, t), \ldots, F_s(p, y, t))^T$ and the initial value $y_0(p)$ are assumed to be continuously differentiable with respect to $p$. Furthermore, $F(p, y, t)$ is assumed to be continuously differentiable with respect to $y$. These two assumptions must be effective on $t$ for all $0 \leq t \leq T$, where $T$ is the terminal time. And they must be valid for all $y$, at
least in the neighborhood of the solution. As pointed out in the following theorem, the derivatives are obtained as solutions to the extended system of non-homogeneous linear differential equations.

**Theorem 3.1.** The solution $y(p, y_0(p), t)$ of the system of differential equations (12) exists and has a derivative with respect to $p$. The derivative is continuous and satisfies the linear matrix differential equation

$$
\dot{D}(t) = \partial_y F(p, y(y_0(p), t), t) D(t) + \partial_p F(p, y(p, y_0(p), t), t),
$$

where $D(t) \in \mathbb{R}^{s \times n}$, with initial value $D(0) = \partial_p y_0(p) \in \mathbb{R}^{s \times n}$.

In this context, $\partial_p$ denotes the partial differential operator dependent on $p$. The Jacobian matrices are

$$
\partial_p F(p, y, t) = \left( \frac{\partial}{\partial p_j} F_i(p, y, t) \right)_{j=1,\ldots,n, i=1,\ldots,s} \in \mathbb{R}^{s \times n}
$$

and

$$
\partial_y F(p, y, t) = \left( \frac{\partial}{\partial y_j} F_i(p, y, t) \right)_{j=1,\ldots,s, i=1,\ldots,s} \in \mathbb{R}^{s \times s}
$$

**Proof.** See [6]. The direct motivation of the sensitive equation (13) is satisfied when comparing (12) with respect to $p$ and exchanging the differentials since the differential equation must be valid for all $t$ and $p$. Set now $D(t; p) = \frac{d}{dp} y(p, y_0(p), t)$, then we can obtain

$$
\dot{D}(t; p) = \frac{d}{dp} \dot{y}(p, y_0(p), t)
$$

$$
= \frac{d}{dp} F(p, y(p, y_0(p), t), t)
$$

$$
= \partial_p F(p, y(p, y_0(p), t), t) + \partial_y F(p, y(p, y_0(p), t), t) D(t; p)
$$

Because we get $y(p, y_0(p), 0) = y_0(p)$ by definition, we can obtain $D(0; p) = \partial_p y_0(p)$. In the equation, $\frac{d}{dp}$ means the total derivative of the subsequent function and $\partial_p$ defines the partial derivatives with respect to $p$. [2]

3.3 Computation of the derivative

In order to solve the least squares problem, the first order derivative of the parameter to be estimated for the fitted function is required [2], also

$$
\frac{d}{d\beta} J(\beta, \gamma) = \frac{1}{N} \sum_{k=1}^{N} (i(\beta, \gamma, t_k) - \tilde{i}_k) \cdot \frac{di}{d\beta} (\beta, \gamma, t_k).
$$

9
For calculating the solution of these ordinary differential equations, sensitivity equations of (6) will be solved. [2]

\[ \frac{\partial}{\partial t} \frac{ds}{d\beta} = -is - \beta \frac{di}{d\beta} s - \beta i \frac{ds}{d\beta}, \]
\[ \frac{\partial}{\partial t} \frac{di}{d\beta} = is + \beta \frac{di}{d\beta} s + \beta i \frac{ds}{d\beta} - \gamma \frac{di}{d\beta} \]  

(16)

with \( i \) and \( s \) are known. This equation system is applicable to the SIR model. Meanwhile, at time point \( t = 0 \), extra start conditions for the equation system are set

\[ i^\beta(0) = 0, \quad s^\beta(0) = 0, \]

as \( s(0) \) and \( i(0) \) are independent of \( \beta \).

### 3.4 Discretization

As we know, the sensitivity equations in the model are infinite-dimensional. For numerically solving the equations, an explicit Euler method is used. [2] The time interval \([a, b]\) will be discretized into \( n + 1 \) points and the equations will then be solved using \( n \) of these points. These discretized time points will be set to (as in Section 2.2)

\[ t_k = a + \frac{b-a}{n} k, \quad k = 0, 1, 2, \ldots, n. \]

Approximate solutions to the equations at these time points will be obtained by an explicit Euler step

\[ s^\beta_{k+1} = s^\beta_k + \frac{b-a}{n} f_s(i_k, s_k, i^\beta_k, s^\beta_k), \quad k = 1, 2, 3, \ldots, n \]
\[ i^\beta_{k+1} = i^\beta_k + \frac{b-a}{n} f_i(i_k, s_k, i^\beta_k, s^\beta_k), \quad k = 1, 2, 3, \ldots, n \]

where the \( f_s \) and \( f_i \) functions are described by the right-hand side of the equations, which means

\[ \dot{i}^\beta(t) = f_i(i, s, i^\beta, s^\beta) = -is - \beta i^\beta s - \beta i s^\beta, \]
\[ \dot{s}^\beta(t) = f_s(i, s, i^\beta, s^\beta) = is + \beta i^\beta s + \beta i s^\beta - \gamma i^\beta. \]

### 3.5 Piecewise parameter estimation

In this subsection, we will concentrate on estimating the piecewise constant parameter \( \beta \). To explore this problem simply and clearly, we will choose a time interval \( T \) and split it into two equal parts. The piecewise parameter \( \beta \) to be estimated is denoted as \( \beta_1 \) in the first interval and \( \beta_2 \) in the second interval as below.

\[ \beta(t) = \begin{cases} \beta_1 & \text{for } t \in \left[0, \frac{T}{2}\right] \\ \beta_2 & \text{for } t \in \left[\frac{T}{2}, T\right] \end{cases} \]
The least squares optimization here is related to $\beta_1$ and $\beta_2$,

$$J(\beta_1, \beta_2) = \min_{(\beta_1, \beta_2)} \frac{1}{2N} \sum_{k=1}^{N} (i(t_k; \beta_1, \beta_2) - \bar{i}_k)^2$$  \hspace{1cm} (17)

with

$$\dot{s} = -\beta(t)s, \hspace{1cm} \dot{i} = \beta(t)i - \gamma i. \hspace{1cm} (18)$$

To clearly describe the process, we first define

$$J^{\beta_i} := \frac{d}{d\beta_i} J$$

for $i = 1, 2$. The derivative of $J$ here is

$$J^{\beta_i} = \frac{1}{N} \sum_{k=1}^{N} (i(t_k) - \bar{i}_k) \cdot i^{\beta_i} \quad \text{for } i = 1, 2 \hspace{2cm} (20)$$

also with the necessary condition

$$J^{\beta_i} = 0 \quad \text{for } i = 1, 2, \hspace{1cm} (21)$$

where $\beta_a \leq \beta_i \leq \beta_b$.

To solve this problem, we need sensitivity equations for $\beta_1$ and $\beta_2$. First, we discuss the sensitivity equations for $\beta_1$. In the time interval $[0, T/2]$, the equations are similar to (16). Due to $i^{\beta_1} \neq 0$ in the first interval, $i^{\beta_1}$ will still remain positive for a while after $t = T/2$. Therefore, the parameter $\beta_1$ also has an effect on the number of infections in the second interval. Similarly, the parameter $\beta_1$ impacts the number of susceptible in the time interval $[T/2, T]$. The contact rate $\beta(t)$ here is time-dependent.

$$t \in [0, \frac{T}{2}] : \quad \partial_t s^{\beta_1} = -is - \beta(t)i^{\beta_1}s - \beta(t)is^{\beta_1}$$

$$= -is - \beta_1i^{\beta_1}s - \beta_1is^{\beta_1},$$

$$\partial_t i^{\beta_1} = is + \beta(t)i^{\beta_1}s + \beta(t)is^{\beta_1} - \gamma i^{\beta_1}$$

$$= is + \beta_1i^{\beta_1}s + \beta_1is^{\beta_1} - \gamma i^{\beta_1},$$

$$t \in \left[\frac{T}{2}, T\right] : \quad \partial_t s^{\beta_1} = -\beta_2i^{\beta_1}s - \beta_2is^{\beta_1},$$

$$\partial_t i^{\beta_1} = \beta_2i^{\beta_1}s + \beta_2is^{\beta_1} - \gamma i^{\beta_1}.$$ 

Next, we come to $\beta_2$. For $t \in [0, T/2]$ it is

$$\partial_t s(t) = -\beta_1i(t)s(t)$$

and then for $\beta_2$ the sensitivity equations is

$$\partial_t s^{\beta_2} = -\beta_1i^{\beta_2}s - \beta_1is^{\beta_2},$$

$$\partial_t i^{\beta_2} = \beta_1i^{\beta_2}s + \beta_1is^{\beta_2} - \gamma i^{\beta_2}.$$
But \( i(t), s(t) \) do not depend on \( \beta_2 \) in this interval, i.e.

\[
i^{\beta_2} \equiv 0, \quad s^{\beta_2} \equiv 0.
\]

For \( t \in (T/2, T] \), \( \beta_2 \) effects on \( s(t), i(t) \), so the sensitivity equations are

\[
\begin{align*}
\partial_t s^{\beta_2} &= -is - \beta_2 i^{\beta_2} s - \beta_2 i s^{\beta_2}, \\
\partial_t i^{\beta_2} &= is + \beta_2 i^{\beta_2} s + \beta_2 i s^{\beta_2} - \gamma i^{\beta_2}.
\end{align*}
\]

### 3.6 Algorithm

For a better understanding of the whole process, i.e. the parameter estimation, for necessary algorithms are shown below.

The first algorithm is to solve the sensitivity equation using explicit Euler's method. The algorithm starts at the initial values \( i_0^\beta \) and \( s_0^\beta \in \mathbb{R} \). In each loop (steps 7 to 10), \( T/n \) times the value of right-hand side of the sensitivity equation (16) at time point \( t \) in point \( (s_k^\beta, i_k^\beta) \) will be added until the number of iterations \( k \) equals the number of fineness \( n \). The second algorithm is the gradient descent algorithm. We start with the initial value \( J_0^\beta \in \mathbb{R}_+ \) and the parameter \( \beta_0 \in \mathbb{R}_+ \). In the loop of this algorithm, we first compute \( s_k(t), i_k(t) \) from the SIR model of (7) by the explicit Euler method. Algorithm 1 is then used to find the approximation of the final value \( i_n^\beta, s_n^\beta \) to \( i^\beta(T), s^\beta(T) \) at each time point. We then have sufficient data to calculate the value of \( J_k^\beta \). Following the gradient method mentioned previously, we set the direction of descent \( d = -\nabla J^\beta \). The final \( \beta_k \) is determined by iterating over the \( \beta_k \). The stopping criterion here is based on the absolute value of \( J_k^\beta \). The loop is interrupted when the absolute value is less than the tolerance. The following two main algorithms are shown to estimate the piecewise constant \( \beta \). Algorithm 3 describes the process of obtaining the final values \( i^\beta(T), s^\beta(T) \), \( i^\beta(T), s^\beta(T) \) of piecewise parameter \( \beta \) in two identical time intervals by explicit Euler method. The initial values in the algorithm are \( i_0^\beta, s_0^\beta \), \( i_0^\beta, s_0^\beta \). In the time interval \([0, T/2] \), \( f_1 \) containing the sensitivity equations will be used for the explicit Euler method, which means that the sensitivity equations are added to \( y_k \) for \( T/n \) times in each loop until the number of iterations \( k \) reaches \( n/2 \). The approximation \( y_{n/2} \) is thus obtained by iteration. Then when the number of iteration \( k \) goes from \( n/2 \) to \( n \), i.e., in the time interval \([T/2, T] \), we obtain the final value through iterations of \( f_2 \) containing the sensitivity equation by adding \( T/n \) times to \( y_k \). As for Algorithm 4, it explains the effect of the gradient descent method on the piecewise parameter \( \beta \). A loop will be set up first. In this loop, an explicit Euler method in SIR-model is used to find the values of \( s_k(t), i_k(t) \) for all times and algorithm 3 to obtain the values of \( s^\beta_k(t), i^\beta_k(t), s^\beta_k(t), i^\beta_k(t) \) in the whole time. Then with these values, we calculate the value of \( J_k^\beta \). Next we use \( J_k^\beta \) to determine \( \beta_{i,k+1} \) by the gradient descent method. Through iterations of \( \beta_{i,k+1} \), \( J_k^\beta \) changes. Until \( J_k^\beta \) is below the given tolerance, the loop ends and the final value \( J_k^\beta \) and the final parameter \( \beta_{i,k} \) will be acquired.
Input: Initial values \(i^\beta_0, s^\beta_0 \in \mathbb{R}\), Parameter \(\beta, \gamma \in \mathbb{R}_+\)

Fineness \(n \in \mathbb{N}\), End time \(T \in \mathbb{R}_+\), Function \(i, s\)

Output: Approximations \(i^\beta_n, s^\beta_n\) to final values \(i^\beta(T), s^\beta(T)\)

Set \(k = 0\);

Set \(y_k = (s^\beta_k, i^\beta_k)\);

Set \(f = (f^s, f^i) = (-is - \beta i s^\beta, is + \beta is)\);

while \(k < n\)

Set \(y_{k+1} = y_k + T/n \cdot f(y_k)\);

Set \(k = k + 1;\)

end

Algorithm 1: sensitivity equations using explicit Euler method

Input: Initial values \(J^\beta_0 \in \mathbb{R}_+, \) Initial parameter \(\beta_0 \in \mathbb{R}_+,\)

Parameter \(\gamma \in \mathbb{R}_+, \) stepsize \(\alpha \in \mathbb{R}_+,\)

Fineness \(N \in \mathbb{N}, \) End time \(T \in \mathbb{R}_+\)

Output: Final value \(J^\beta_k \in \mathbb{R}_+, \) Final parameter \(\beta_k \in \mathbb{R}_+\)

Set \(k = 0;\)

while \(|J^\beta_k| < \epsilon\)

Determine \(s_k(t), i_k(t)\) from SIR-model, \(t = 0, \ldots, T;\)

Determine \(s^\beta_k(t), i^\beta_k(t)\) from Algorithm 1, \(t = 0, \ldots, T;\)

Determine \(J_k^\beta = 1/N \cdot \sum_{m=1}^N (i_m - i_m^\beta) i_m^\beta;\)

Set \(\beta_{k+1} = \beta_k - \alpha \cdot J_k^\beta;\)

Set \(k = k + 1;\)

end

Algorithm 2: gradient descent algorithm
Algorithm 3: sensitivity equations for piecewise parameter using explicit Euler method

Algorithm 4: gradient descent algorithm for piecewise parameter

4 Results

All results have been obtained using Python. We set the recovery rate $\gamma = 0.13$ to be a contact function. [4] As for the number of intervals $n$, we first set it as 100. The tolerance $\epsilon$ is first set to 0.0001.
4.1 Example 1: Manufactured solution

In Figure 2, we first assume a parameter $\bar{\beta} = 0.3$ and then find the proportion of currently infected people in the total population $i(\bar{\beta})$ accordingly, which is the manufactured function, represented by the red line. Certainly, there has to be an initial parameter $\beta_0$ with which to attempt to find the final fitted curve. The blue line indicates the proportion of infected persons at $\beta_0$. As we can see from the graph, there is a lot of distance between these two lines that we need to shorten. We will approach the manufactured line by changing $\beta$. And the change in $\beta$ is related to stepsize $\alpha$, as different $\alpha$ will lead to a different process of change in $\beta$. The larger $\alpha$ is, the more drastic the change in $\beta$.

Effect of the stepsize alpha Therefore, we tried to observe the relationship between the stepsize $\alpha$ and the infectious functions, which is related to different variations of $\beta$, and then choose a suitable $\alpha$. When $\alpha \geq 0.5$, $\beta$ will first go to a low value and $J^\beta$ will first be a large negative number. Then through iteration, $\beta$ will gradually get larger to approach $\bar{\beta}$, and $J^\beta$ becomes progressively smaller, which means the changing infection function will in turn approach the manufactured function until the loop break condition is met. This implies that the reduction in $\beta$ at the start is too great. In the case of $\alpha \leq 0.5$, the starting function approaches the manufactured function positively, but when $\alpha = 0.1$, the iteration number is 103, it takes too many steps. And at $\alpha = 0.3$ the number of iterations only takes 28. Hence in the end $\alpha = 0.3$ will be the decision to choose. These illustrate that a smaller $\alpha$ leads to a larger number of iterations because the value of $\alpha$ affects the reduction of $\beta$ in each iteration. The optimal value of $\beta$ with respect to $\alpha = 0.3$ during the iterative process is 0.3199. The right column shows the time curve of the susceptible number. The starting function will first run to higher and then descend when $\alpha \geq 0.5$. As for $\alpha \leq 0.5$, $\alpha = 0.1$ requires more iterative steps than $\alpha = 0.3$, even though they end up with similar susceptible functions, which are manufactured functions.

Variation of the time-step size Delta t We next want to explore the relationship between the time curves and the time intervals. To keep all other conditions constant, we will set $\alpha = 0.3$. As shown in Figure 3, we choose $\Delta t = 1/2, 1/5, 1/10, 1/20$, also $n = 20, 50, 100, 200$ to compare the related time curves. As in the above figure, the blue line indicates the starting function and the red line indicates the manufactured function, which is what we need to achieve. First, we focus on the $\Delta t = 1/20$ case. The time curves in all iterations are lower than the manufactured function. This is because the first iteration of $\beta$ leads to a huge drop, which means that a large time interval will compute more data and then we get a huge $J^\beta$. The optimal $\beta$ for $\Delta t = 1/20$ is 0.3201. Then we concentrate on $\Delta t = 1/2$, we can observe that the pink lines consistently appear in the middle of the blue and red lines. It took 104 iterations to get the final $\beta$ 0.3185. This shows that a large time-step size leads to more iterations due to the value of $J^\beta$. When $\Delta t = 1/5$, the number of iterations is reduced to 57 and the final $\beta$ is 0.3193. The optimal case is when $\Delta t = 1/10$, which costs only 28 iterations, and the final $\beta$ is 0.3199.
Then let us look at the right column, where these four images show the corresponding susceptible functions. When $\Delta t = 1/20$, the time curve of the susceptible function will rise and then fall back. When comparing $\Delta t = 1/2, 1/5, 1/10$, we find that a smaller time-step size leads to fewer iterations.

**Semi-log Graph** We use parameter $\alpha = 0.3, \Delta t = 1/10$ to set the semi-log graph. From the Figure 4, we know that the drop in $J$ is rapid in the early iterations and then slows down sharply in the later ones, while the drop in $J^\beta$ is rapid at the beginning and then exponential from iteration 2 to the end.
Figure 2: Time curves under different step sizes
Figure 3: Time curves of different time step sizes
4.2 Example 2: Real data

Real data will be applied next to estimate parameter $\beta$. We choose the 7-day average COVID-19 data in Germany from 1 January to 31 January for Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) COVID-19 dashboard in Table 1. [5]. The JHU CSSE will offer instant global data on COVID-19. As for the data from Germany, JHU CSSE’s data is always available earlier than from many German sources. After obtaining the data, we calculate the 7-day average. The 7-day average was chosen to reduce the impact of delays in data registration and to improve the reference value of the data. As with the manufactured solution, we will first observe which stepsize is more suitable for the real data. In the case of $\alpha \geq 4$, $\beta$ drops to a lower value and then increase to be closer to the real case. But what we need is a step down from the initial function to the tracking function, so $\alpha$ will be chosen to be less than 4. At $\alpha = 2$, the number of iterations reaches 109, which means too much. The iteration steps are shown as 68 when $\alpha = 3$, which is appropriate. To better approximate the tracking function, the stepsize $\alpha = 3$ is set here. The optimal value for $\beta$ is equal to 0.1911. Our goal is to bring the starting function closer to the tracking function, Figure 5a illustrates this process. In order to better observe the results, we set up Figure 5b here, the last
found infected function is set as the green line and the real data as the red dots. Figure 5b shows the results in more detail and more clearly, and we can realize that the function is close. Certainly, we can notice that the curve fits well in the beginning, but grows too fast at the end. This means that we need a smaller $\beta$ here, so the piecewise constant $\beta$ is needed. In Figure 6, the semi-log function of $J$ shows a sharp drop to a slow drop, while from the semi-log function of $J^\beta$ we can see a uniform fast exponential drop at the beginning and a continuous uniform slower exponential drop afterward.
(a) Time curve of infectious function

(b) Real data vs. calculated data

Figure 5: Real data fitting
4.3 Example 3: Manufactured solution for piecewise parameter

With Algorithm 3 and 4, we can manufacture an ideal case of the infection number time curve for the piecewise parameter $\beta$. We set the end time $T = 10$ and the turning point $T_m = 5$, then there will be two equal time intervals of equal length $[0, 5]$ and $(5, 10]$. The initial piecewise parameters $\beta_1$ and $\beta_2$ are then both set to 0.3, which also leads to the initial time curve, which is blue. Then we assume that the piecewise parameters $\bar{\beta}_1 = 0.5$ and $\bar{\beta}_2 = 0.1$, the goal values $\tilde{i}_k$ in the functional 20 are computed based on them. As shown in Figure 7, the time curve will start from starting function and then the curve will have a turning point at $t = 5$. This is because $\beta_1$ will gradually get larger during the iterations until it approaches $\bar{\beta}_1$, and $\beta_2$ will gradually get smaller until it approaches $\bar{\beta}_2$, at the meanwhile the curve will gradually approach the manufactured function until they almost overlap. The parameter $\beta_1$ and $\beta_2$ are optimized simultaneously during the process. We selected some steps $\alpha$ for comparison. The number of iterations reaches 148 when $\alpha = 0.1$, while $\beta_1 = 0.4987$ and $\beta_2 = 0.1029$. When $\alpha = 0.3$, the iteration number reduce to 49 with $\beta_1 = 0.4989$ and $\beta_2 = 0.1026$. Iteration numbers becomes 37 with $\beta_1 = 0.4990$ and $\beta_2 = 0.1023$ when $\alpha = 0.4$. At time $\alpha = 0.5$, $\beta_1 = 0.4990$ and
$\beta_2 = 0.1020$ after 30 iterations. For $\alpha \geq 0.5$, $\beta$ has huge changes during the iteration and the stopping criterion is never satisfied, so the loop does not end. By comparison, $\alpha = 0.4$ is a good stepsize because the time curve changes significantly and the number of iterations is short enough. We continue find out the curve of $J$, $J^{\beta_1}$ and $J^{\beta_2}$. As shown in Figure 8b, the $J$ in red is a straight line descending at a uniform velocity, which indicates that it is decreasing exponentially. Moving on to the blue line, since $J^{\beta_1}$ is negative, we have plotted regularly for it. The value of $J^{\beta_1}$ will start at -0.0016 and then quickly rise to -0.0002 and then slowly climb until it approaches zero. And the semi-log plot of $J^{\beta_2}$ in purple on the right side grows sharply larger at first, then decreases exponentially.

Figure 7: Time curve for piecewise $\beta$
Figure 8: Semi-log plot of $J$, $J^{\beta_1}$ and plot of $J^{\beta_1}$
4.4 Example 4: Real data for piecewise parameter

For real data, we expected the infection curve in a time period to include a turning point, similar to our manufactured function. By viewing the JHU CSSE COVID-19 dashboard [5], we choose the data for the period January 18, 2022 to February 27, 2022, and the applied data are shown in Table 1. First, we download the number of infectious cases for this period and then calculate the 7-day average and apply it to our graphs. The 7-day average data reduced errors due to reporting delays, which then improved the curve fit. Figure 9a contains two plots, where the difference is the time at which the function’s turning point is located. On the green plot, we choose the turning point $T_m = 20$, which is in the middle of the whole period. On the yellow graph, the turning point is $T_m = 17$. Both intervals $[0, 17]$ and $[17, 40]$ have the same size of the sub-interval $n$. It is clear that the infection function in both plots starts with the starting function and then approaches the tracking function step by step, which is connected by real data. The initial piecewise parameters to be estimated, $\beta_1$ and $\beta_2$, will both be set to 0.15. The stepsize, $\alpha$, will be set to 30. During the iterations, $\beta_1$ will become larger and $\beta_2$ will gradually become smaller. The process in the left graph took 61 iterations, and finally $\beta_1 = 0.1727$ and $\beta_2 = 0.1149$. As for the graph on the right, the number of iterations is 51, and the final parameters become $\beta_1 = 0.1782$ and $\beta_2 = 0.1199$. As for the values $J(\beta_1, \beta_2)$ for the optimal values $\beta$, it is 2.9823e-07 in the first case and 1.0988e-07 in the second case. From this value, it can be seen that the curve fits better in the second case.

We can then gain a better insight into the function fit from Figure 9b, where the red dots are the real data plotted, the green line is the last infected curve when the turning point is 20, and the yellow line is the last infected image when the turning point is 17. It is noticeable that there is still a huge space between the red dot and the green line, where the green line is first substantially lower than the red dot, but continues to be higher than the red dot in the middle section. The yellow line, on the other hand, fits the data relatively well, with the turning point right in the middle of the maximum change. Also, since the effect about $\beta_1$ will remain in the second time period, the turning point a little earlier than the highest point will obtain the best fitting function. Therefore, we plot the semi-log plots of the values of $J$, $J^{\beta_2}$, and plot of $J^{\beta_1}$ for the case where the turning point is $T_m = 17$. From the graph 10a, the semi-log plot of $J$ first decreases in a straight line, which means that the rate of decline of $J$ increases exponentially. Then, the decreasing trend slows down until it is constant. Moving on to Figure 10b, We can see that the value of $J^{\beta_1}$ starts at -6, increases sharply over time with a few sharp fluctuations, and then slows down until it stabilizes at zero. The semi-log plot of $J^{\beta_1}$, meanwhile, starts with a little increase, followed by a few fluctuations, and then keeps going straight down.
(a) Time curve of infectious function for piecewise $\beta$ in real data

(b) Real data vs. calculated data for piecewise $\beta$

Figure 9: Real data fitting for piecewise $\beta$
Figure 10: Semi-log plot of $J$, $J^{β_2}$ and plot of $J^{β_1}$ in real data
5 Conclusion and Summary

In this thesis, the SIR model is first constructed and the algorithm for estimating the contact rate $\beta$ in the SIR model is derived. In this process, the explicit Euler method is applied to the sensitivity equation in order to calculate the first-order derivative of the given function. Numerical results for manufactured solutions have been presented and fitted to the real COVID-19 data from Germany. The gradient algorithm which was used works well only for an appropriate stepsize $\alpha$. A too large stepsize can cause a huge drop on the curve in the first iteration and a too-small one will result in slow convergence. Certainly, proper time-step size is also required. As for the parameters being estimated, we first discuss constant contact rates. Since the curve fitting of real data can still be improved, we consider the estimation of the piecewise constant parameter. It is significant that a good fit can be obtained by capturing the turning points correctly when estimating the piecewise constant parameters. Mathematical models can be considered as an effective tool for investigating disease outbreaks. At the same time, this model could be improved in the following points. In the gradient descent method, using the Armijo line search method will give better results. For the estimation of parameters, $\gamma$ could also be considered. It is known that SIR-like models are widely used to model the spread of epidemics, such as the SEIR model [10]. There are some other parameters in these models. We can also construct these models and then estimate the parameters. Another drawback concerns the estimation of piecewise parameters. A better estimate of $\beta(t)$ is obtained when the value of beta varies over more than 2 intervals or in some extreme cases, even when the estimate of $\beta(t)$ is a continuous function.
References


29
<table>
<thead>
<tr>
<th>Date</th>
<th>Infectious Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/22</td>
<td>30291.29</td>
</tr>
<tr>
<td>1/2/22</td>
<td>32986.86</td>
</tr>
<tr>
<td>1/3/22</td>
<td>36068.29</td>
</tr>
<tr>
<td>1/4/22</td>
<td>38224.71</td>
</tr>
<tr>
<td>1/5/22</td>
<td>42438.57</td>
</tr>
<tr>
<td>1/6/22</td>
<td>45872.43</td>
</tr>
<tr>
<td>1/7/22</td>
<td>46834.86</td>
</tr>
<tr>
<td>1/8/22</td>
<td>48996.14</td>
</tr>
<tr>
<td>1/9/22</td>
<td>52070.14</td>
</tr>
<tr>
<td>1/10/22</td>
<td>54509.71</td>
</tr>
<tr>
<td>1/11/22</td>
<td>59636.57</td>
</tr>
<tr>
<td>1/12/22</td>
<td>62798.43</td>
</tr>
<tr>
<td>1/13/22</td>
<td>65077.29</td>
</tr>
<tr>
<td>1/14/22</td>
<td>66347.29</td>
</tr>
<tr>
<td>1/15/22</td>
<td>70449.43</td>
</tr>
<tr>
<td>1/16/22</td>
<td>75005.57</td>
</tr>
<tr>
<td>1/17/22</td>
<td>82451.14</td>
</tr>
<tr>
<td>1/18/22</td>
<td>89299.29</td>
</tr>
<tr>
<td>1/19/22</td>
<td>97504.86</td>
</tr>
<tr>
<td>1/20/22</td>
<td>102210.00</td>
</tr>
<tr>
<td>1/21/22</td>
<td>106388.29</td>
</tr>
<tr>
<td>1/22/22</td>
<td>113895.43</td>
</tr>
<tr>
<td>1/23/22</td>
<td>121277.86</td>
</tr>
<tr>
<td>1/24/22</td>
<td>131220.71</td>
</tr>
<tr>
<td>1/25/22</td>
<td>138361.86</td>
</tr>
<tr>
<td>1/26/22</td>
<td>146034.00</td>
</tr>
<tr>
<td>1/27/22</td>
<td>150824.00</td>
</tr>
<tr>
<td>1/28/22</td>
<td>152956.14</td>
</tr>
<tr>
<td>1/29/22</td>
<td>158050.14</td>
</tr>
<tr>
<td>1/30/22</td>
<td>164407.00</td>
</tr>
<tr>
<td>1/31/22</td>
<td>169119.00</td>
</tr>
<tr>
<td>2/1/22</td>
<td>177503.29</td>
</tr>
<tr>
<td>2/2/22</td>
<td>181596.00</td>
</tr>
<tr>
<td>2/3/22</td>
<td>183625.00</td>
</tr>
<tr>
<td>2/4/22</td>
<td>186046.29</td>
</tr>
<tr>
<td>2/5/22</td>
<td>187040.29</td>
</tr>
<tr>
<td>2/6/22</td>
<td>190719.14</td>
</tr>
<tr>
<td>2/7/22</td>
<td>192396.57</td>
</tr>
<tr>
<td>2/8/22</td>
<td>191158.57</td>
</tr>
<tr>
<td>2/9/22</td>
<td>190012.00</td>
</tr>
<tr>
<td>2/10/22</td>
<td>188867.29</td>
</tr>
<tr>
<td>2/11/22</td>
<td>186181.29</td>
</tr>
<tr>
<td>2/12/22</td>
<td>184702.14</td>
</tr>
<tr>
<td>2/13/22</td>
<td>182662.43</td>
</tr>
<tr>
<td>2/14/22</td>
<td>180914.43</td>
</tr>
<tr>
<td>2/15/22</td>
<td>178039.57</td>
</tr>
<tr>
<td>2/16/22</td>
<td>175084.71</td>
</tr>
<tr>
<td>2/17/22</td>
<td>174066.43</td>
</tr>
<tr>
<td>2/18/22</td>
<td>173695.29</td>
</tr>
<tr>
<td>2/19/22</td>
<td>168936.00</td>
</tr>
<tr>
<td>2/20/22</td>
<td>167376.00</td>
</tr>
<tr>
<td>2/21/22</td>
<td>164618.29</td>
</tr>
<tr>
<td>2/22/22</td>
<td>163289.00</td>
</tr>
<tr>
<td>2/23/22</td>
<td>161393.00</td>
</tr>
<tr>
<td>2/24/22</td>
<td>159947.43</td>
</tr>
<tr>
<td>2/25/22</td>
<td>158302.00</td>
</tr>
<tr>
<td>2/26/22</td>
<td>157760.43</td>
</tr>
<tr>
<td>2/27/22</td>
<td>154525.29</td>
</tr>
</tbody>
</table>

Table 1: Number of infections by Data from JHU CSSE