University of Leiden
Mathematical Institute
Bachelor Thesis

Modelling the Role of Lockdowns in Reducing the Spread of Infectious Diseases

Esther Stehouwer

Supervisor:	Dr. F.W.J. Veerman
Date final version:	30-06-2023

Abstract

Background: COVID-19 was the first infectious disease to be declared as a pandemic in history. With over 7 million death due to the consequences of COVID-19, and significant impact on the economy, healthcare systems, and social life, the virus can be considered one of the most serious of our time. It is likely that more pandemics will occur, due to globalisation. Many factors exist that can influence the pandemic (e.g., pharmaceutical and social interventions). One factor that can reduce the influence of a pandemic is implementing a lockdown. More knowledge on the effectivity of lockdowns during COVID-19 could increase our knowledge on how to implement a lockdown during a second pandemic.

Objectives: Currently, there is still little data on whether a lockdown as a measure to reduce the spread of a virus is effective. Therefore, the objective of this study was to determine the effectivity of the lockdown in the Netherlands during the period from 01-03-2020 to 01-06-2020, to analyze whether there could have been a more optimal starting day or duration, and to examine whether the lockdown was Pareto Efficient.

The Effectiveness of a Lockdown: The effectiveness of the lockdown was determined by simulating the spread of the infection of COVID-19 using the SIR- (Susceptible, Infectious, Recovered) and SEIRS- (Susceptible, Exposed, Infectious, Recovered, Susceptible) models with and without lockdown, and comparing the maximum infection value peak with each other. The SIR-model showed, after parameter fitting, a fair simulation of the infection curve in the Netherlands, and an lockdown effectiveness of 40.5 %. The SEIRS-model showed, after parameter fitting, a poor simulation of the infection curve in the Netherlands, and a lockdown effectiveness of 0.004%. Thus, in the SIR-model, the lockdown could be considered effective as a measure to decrease the spread of COVID-19 at that time, whereas, in the SEIRS-model, it could not.

Optimal Starting Day and Duration: A more optimal starting day and duration than the lockdown implemented, which was from 15-03-2020 until 15-05-2020, was determined by plotting three-dimensional and contour plots using the SIR- and SEIRS-models of all possible starting days and durations. For the SIR-model, the plots showed that if the starting day would have been four days later, but the duration would have stayed 57 days, it would have yielded an effectiveness of 79.1 %. Changing the duration from 57 days to 49 days would have yielded an effectiveness of 71.2 %. For the SEIRS-model, the plots showed that the starting day and duration were already optimal and that changing one of either would not increase the effectivity of the lockdown, and only worsen it. Thus, a correct moment of implementing a lockdown for a correct period of time was shown to have a determining impact on the infection values.

Pareto Efficiency: Pareto efficiency is defined as a situation in which it is not possible to optimize the one situation without it being at the expense of another situation. In this case, the two objectives being optimised are the duration of the lockdown multiplied by the lockdown strength on the one hand, versus the maximum infection value on the other hand. The Pareto Front, which is defined as the set of all Pareto efficient solutions, were plotted for both the SIR- and SEIRS-model. The lockdowns in both the SIR- as well as the SEIRS-models were shown not to be Pareto efficient.

Conclusion: Implementing a lockdown as a measure to limit the spread of an infectious disease is effective in the SIR-model, but not effective in the SEIRS-model when using COVID-19 data obtained from the World Health Organization, for the parameter values used in these models. However, effectiveness of the lockdown is strongly dependent on correct parameter values, since simulations of the SIR- and SEIRS-model with arbitrary parameter values showed that implementing a lockdown is very effective. Therefore, more research should be

done on the parameter values that are specific for COVID-19. Then, a correct moment of implementing a lockdown for a correct period of time has an important impact on infection values, and, in the SIR-model, the lockdown could have been more efficient. Lastly, the lockdowns in the both the SIR- as well as the SEIRS-models were shown not to be Pareto efficient.

Contents

Acknowledgements

First, I am grateful to dr. Frits Veerman, who has been my supervisor for the entire process of completing this bachelor thesis and who has helped me to put this bachelor thesis together. He has helped me come up with the subject of this bachelor thesis and helped me gain more knowledge in the field of data science and measures on the spread of infectious diseases. I am very thankful for the weekly meetings in which I could ask all possible questions which has brought the bachelor thesis to the level where it is now.

Furthermore, I would like to express my gratitude for all academic staff at the Mathematical Institute at Leiden University, who brought me to the academic level where I am now.

Lastly, I would like to thank my friends and family, who supported me during the process of writing this bachelor thesis together with working and rowing on a high competitive level. A special thanks goes to my father, who has always been willing to help me when I needed it and commented on earlier versions of this thesis.

1 Introduction

In 2019, 2.11 million people worldwide died from the consequences of lower respiratory tract infections, which accounted for approximately 4% of all deaths of that year [25]. Lower respiratory infections (LRTI) are one of many infectious diseases, which are illnesses caused by viruses, parasites, bacteria, or fungi. LRTIs specifically refer to infections of the lower airways of the lungs [22]. Even though incidence rates have been decreasing since 1990, the absolute number of LRIs has been increasing ever since because of population growth and ageing [19]. Moreover, diminishing the burden of LRTIs has been shown to be very difficult since many different approaches can be taken to alleviate this burden, such as medical interventions (e.g., vaccinations and medicines), behavioural changes (e.g., improving hygiene and ventilation), and government regulations (e.g., mandatory minimum distance and implementing a lockdown) [17]. Therefore, there is a clear necessity of researching the impact of such measures on the burden of LRTIs.

COVID-19, caused by the SARS-CoV-2 virus, is one of the most serious LRTIs of our time. The virus has spread rapidly across the globe since it was first identified in Wuhan, China in late 2019 [30]. Since December 2019, approximately 7 million people have died due to the consequences of COVID-19 [3]. The virus has had a significant impact on the economy, healthcare systems, and social life in affected countries [13]. Firstly, the virus has had a severe impact on the global economy. According to the Statista Research Department from Germany, the COVID pandemic has caused the worst economic downturn in 2020 since the Great Depression, with a 3.4 % decline in global domestic product (GDP) in 2020 [1]. The COVID pandemic has also been shown to have a severe and abrupt economic impact, since the projected GDP growth initially was 2.9 % growth in 2020 [1]. The pandemic has disrupted global supply chains, reduced demand for goods and services, and caused widespread job losses, particularly in the tourism and hospitality sectors [2].

Secondly, the healthcare systems of many countries have been overwhelmed by the rapid spread of COVID-19. Hospitals have struggled to provide care for critically ill patients, and shortages of personal protective equipment and other medical supplies have made it difficult for healthcare workers to do their jobs safely [12]. The pandemic has also disrupted the provision of other essential health services, such as vaccinations, cancer screenings, and elective surgeries, putting the health of many people at risk [10]).

Thirdly, the social impact of the pandemic has been significant, with people experiencing high levels of stress, anxiety, and social isolation [14]. The pandemic has disrupted education systems, with many schools and universities closing temporarily, and has made it difficult for people to socialize and participate in cultural and religious events [9]. The pandemic has also highlighted existing inequalities in society, with vulnerable populations such as the elderly, the socially deprived, and those with underlying health conditions at increased risk of severe illness and death [28].

The above has shown that the impact of a pandemic can be immense. Additionally, due to an increasing world population and increasing globalisation it is reasonable to expect that a second pandemic may arise. An increase in world population leads to an increase in the number of local populations living in more densely populated areas which in turn increases the chance of viruses being able to spread among people [18]. Globalisation, which can be defined as a

process in which economic structures, different cultures and political systems of countries come to terms with each other and even become dependent on each other, has led to an increase in international trade and travel, which has led to an increase in the number of people that potentially carry a virus with them to another country [7, 26]. Additionally, globalisation has induced a rise in urbanization which leads to more rapid spread of a virus among people [4]. Lastly, globalisation has led to a change in the production and consumption of different kinds of food [11]. This has increased the chance of transmitting pathogens from animals to humans [11]. Thus, globalisation can lead to a quicker spread of a virus across continents, countries, and populations.

Because a pandemic has an immense impact on populations, and because there is a possibility that a second pandemic will arise, research on minimizing the impact of a pandemic on populations is clearly relevant. There are many factors that can play a role in reducing the influence of a pandemic, both at the pharmaceutical level (e.g., vaccines) as on the societal level (e.g., washing hands more often) [5]. One of the many ways that were used to reduce the spread of COVID-19 is by introducing a lockdown. Lockdowns are a public health intervention that involves restricting the movement of people and closing non-essential businesses and services to reduce the spread of an infectious disease [14]. Due to mandatory social distancing in the form of quarantines, closure of schools, churches, and public facilities, and banning of social activities that include mass gatherings such as festivals and protests, the spread of the virus has been significantly reduced [29]. However, the impact of lockdowns on society has been shown to be immense, both in terms of mental health issues (e.g., anxiety disorders and loneliness), and economic impact (e.g., job losses) [2, 14].

This raises the question of how effective lockdowns are as a measure to reduce the spread of a virus, and the related question of how long and how intense a lockdown needs to be to be effective. These questions van be addressed by analysing a lockdown implemented in the past in which the Pareto Front (which will be defined below) and a more optimal starting day and duration of the lockdown at that time are determined. For this thesis, the period from March 2020 until June 2020 was chosen, since the first lockdown was from 15-03-2020 until 11-05-2020 in the Netherlands [31]. In consideration of the above, it is worth examining the following research questions:

- ***What was the effectivity of the lockdown in the Netherlands during the period from 01-03-2020 to 01-06-2020?***
- ***Could the lockdown in the Netherlands during the period from 01-03-2020 to 01-06-2020 have had a more optimal starting day or duration?***
- ***Could the lockdown in the Netherlands during the period from 01-03-2020 to 01-06-2020 be considered Pareto efficient?***

2 The Lockdown to be analyzed

To determine the effectivity of the lockdown in the Netherlands from March 15 until May 11, it is necessary to know the infection curve during that period. Therefore, it was necessary to collect

data containing values of the number of infections from March 2020 until June 2020. The World Health Organization (WHO) possesses open datasets with these infection values for almost all countries in the world [3]. The data that was used for this thesis comes from a dataset called $owid - covid - data.csv$ which was last modified on 02-06-2023. This dataset contains among others the number of new COVID-19 cases per country per day [3]. To model the infection curve of the Netherlands, the columns containing the dates and the new cases of the Netherlands were stored in a new table. To visualize the infection curve during the first lockdown from 15-03-2020 until 11-05-2020 the table was reduced to the period 01-03-2020 to 01-06-2020. Lastly, a new column was created containing the values for the period prevalence, which refers to the number of infected individuals within a certain period, which was ten days in this case, since literature has shown that people are on average infectious for a period of ten days (see Appendix A) [27]. Figure 1 shows a plot of the period prevalence of COVID-19 in the Netherlands from 01-03-2020 until 01-06-2020 with the starting and ending day of the lockdown (15-03-2020 and 11-05-2020).

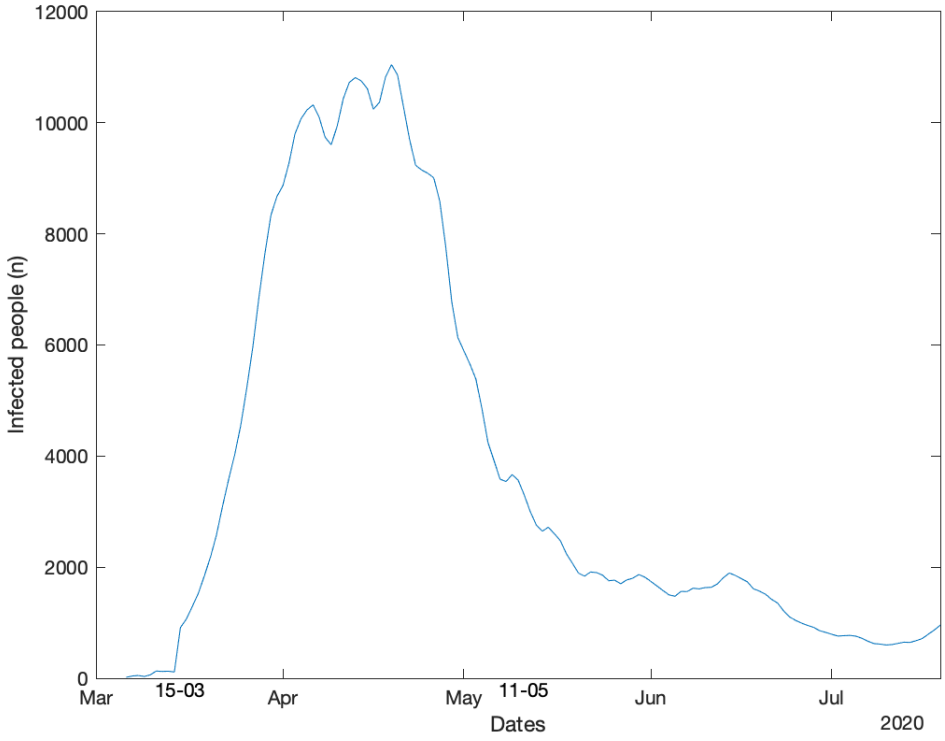

Figure 1: Period prevalence of COVID-19 in the Netherlands from 01-03-2020 to 01-06-2020

3 Theoretical models to determine the effectivity of lockdown

3.1 The SIR-model

3.1.1 Theoretical background

The effectivity of the lockdown can be determined by comparing the maximum infection value without lockdown with the maximum infection value with lockdown. The maximum infection value was specifically chosen, rather than for example the total number of infections over time,

since the main concern during the pandemic was the capacity of available hospital beds and personnel and this is directly related to the maximum infection value. Then, to determine whether the lockdown could have been more optimal, the Pareto Front was determined and a more optimal starting day and duration were determined. However, the period prevalence shown above was not sufficient and the theoretical SIR-model (later SEIRS-model) was needed to determine the Pareto Front and a more optimal starting day and duration of the lockdown. Compartmental models are used to mathematically model an infectious disease and therefore these were used for this thesis [21]. The most generic model is the SIR-model (**S**usceptible, **I**nfectious, **R**ecovered) and was first used to create a model that visualizes the infections over time. The model divides the relevant population into three different compartments: the susceptible compartment consists of people who have not yet been exposed to the virus and when exposed transition into the infectious compartment with rate β, which is the transmission rate; the infectious compartment consists of people who have been exposed to the virus and can infect susceptible people; the recovered compartment consists of people who have recovered from the virus and are considered immune to the virus, which means that they can no longer be infected anymore. The model can be expressed by the following system of ordinary differential equations:

$$\begin{cases} \dfrac{dS}{dt} = -\dfrac{\beta IS}{N}, \\ \dfrac{dI}{dt} = \dfrac{\beta IS}{N} - \gamma I, \\ \dfrac{dR}{dt} = \gamma I. \end{cases} \tag{1}$$

Here, $\frac{dS}{dt}$, $\frac{dI}{dt}$, and $\frac{dR}{dt}$ represent the rate of change in the number of people that are in the S, I or R category respectively with respect to time. The parameters β and γ represent the transmission and recovery rate respectively. N represents the population size. Thus, these equations model the spread of an infectious disease within a population.

For the SIR-model, it is possible to add a cost function $u(t)$ to the model depending on time that represents the lockdown. Assuming that the rate in which the infection is transmitted can be reduced by a factor

$$0 \leq u(t) \leq 1,$$

equation (1) changes into:

$$\begin{cases} \dfrac{dS}{dt} = -\dfrac{(1-u)\beta IS}{N}, \\ \dfrac{dI}{dt} = \dfrac{(1-u)\beta IS}{N} - \gamma I, \\ \dfrac{dR}{dt} = \gamma I. \end{cases} \tag{2}$$

Here, $u(t)$ is in fact a variable that reduces the strength of the parameter β. To start a simulation, the SIR-model, which is one of the simplest compartmental models, can be used to simulate the lockdown that has been implemented in March 2020. It is for this reason that $u(t)$ has also been used first in its simplest form, namely as a constant function: $u(t) = c$ for all $c \in [0, 1]$. For the simulation, $u(t)$ is a stepfunction and is defined as follows (see also Figure 2 and Appendix B):

$$u(t) = \begin{cases} 0, & t \in [0, t_1) \\ c, & t \in [t_1, t_2) \\ 0, & t \in [t_2, \infty) \,, \end{cases} \tag{3}$$

with t_1 the starting day of the lockdown and t_2 the day the lockdown is lifted.

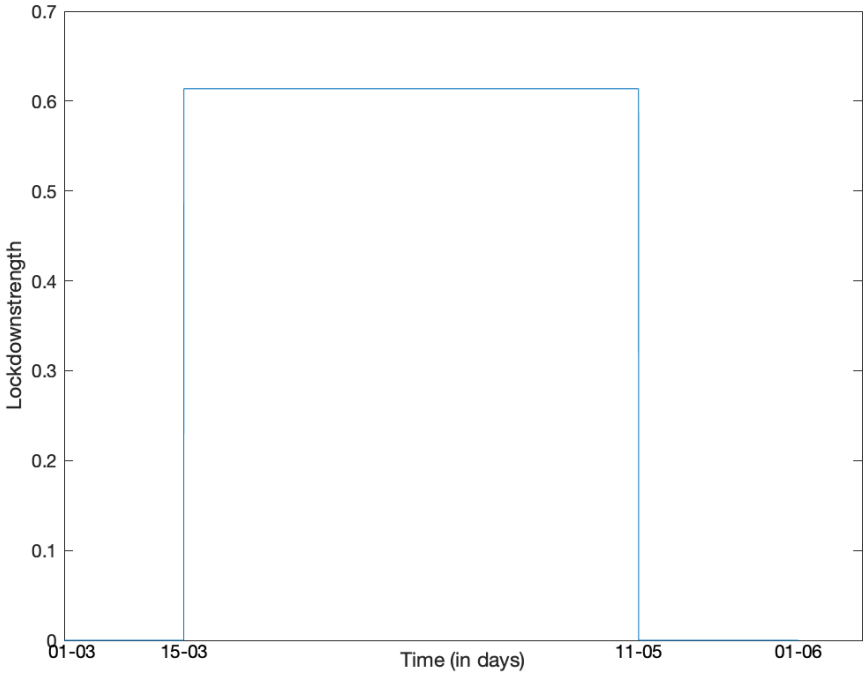

Figure 2: Lockdown strength over time

Figure 3 shows a schematic overview of the SIR-model with the transmission rate β, recovery rate γ, and the cost function $u(t)$.

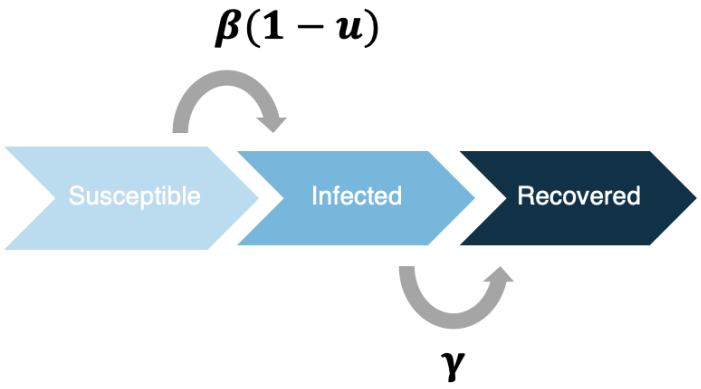

Figure 3: Schematic overview of SIR-model

This model is based on some classical simplifying assumptions: recovery from disease is equal to immunity of disease; the status of vaccinations remains constant over time; the population size remains constant; there is no import of cases from other populations and the population is

homogeneous; and there is no incubation period.

3.1.2 The SIR-model with arbritrary parameter values

Before applying the actual data, it was valuable to analyze the influence of implementing a lockdown function into the SIR-model to obtain some notion on how a lockdown can influence the infection values. This was done by running the SIR-model in MATLAB with parameters values arbitrarily chosen. Table 1 shows an overview of the parameters used for the SIR-model. The lockdown was set on $u(t) = 0.75$ for $t_1 = 20$ and $t_2 = 40$.

Table 1: List of arbitrary parameters used in numerical simulations for SIR-model

Parameter	***Value***	***Meaning***
N	10000000	Population size
t_{max}	100	Length of simulation
S(1)	9999000	Initial value of susceptible people
I(1)	1000	Initial value of infected people
R(1)	0	Initial value of recovered people
γ	0.2	Recovery rate
β	0.6	Transmission rate

Figure 4 shows two infection curves visualising the number of infections over time with and without lockdown (see Appendix C). The curves clearly show the impact of implementing a lockdown, since the maximum infection value decreases when a lockdown is implemented in the model. This makes sense, since the fraction $\frac{\beta IS}{N}$ is multiplied by $1 - 0.75$ which leads to a decrease in the maximum value of this fraction and thus the maximum infection value. Also, the initial value of infected people, $I(1)$, influences the position of the infection curve. The lower the value of $I(1)$, the more the infection curve is positioned to the right, since more time is needed for the infection values to increase. Additionally, multiple simulations of the SIR-model with different arbitrary parameter values has shown that a ratio of $\frac{1}{10000}$ of the initial value of infected people relative to the population size is needed to see the impact of β (and $u(t)$) on the infection curve. Lastly, multiple simulations of the SIR-model when changing arbitrarily parameter values for β and γ, seperately from each other show the following: an increase in β decreases the maximum infection value and the infection curve to start to increase earlier; and an increase in γ decreases the maximum infection value.

Figure 5 shows infection curves for different starting days of a lockdown given a duration of 20 days (see Appendix D). The plots show that for different starting days the lockdown has a different influence on the infection values. For a starting day around the increase in infection values of the original infection curve (blue), there first will be a lower peak and then a higher peak. This makes sense, since the lockdown is implemented during the increase in infection values, which means that the lockdown had less influence during the peak of the lockdown. For a starting day around the peak of the infection curve without lockdown (red and yellow), there will be two peaks at approximately the same height. This also makes sense, since the lockdown is implemented during the peak in infection values, which means that the lockdown had optimal

influence on the maximum infection values. For a starting day around the decrease in infection values of the infection curve without lockdown (purple), there first will be a higher peak and then a lower peak. This again makes sense, since the lockdown is implemented after the peak in infection values is reached, which means that the lockdown had less influence during the peak of the lockdown. Thus, Figure 4 shows that implementing a lockdown can be valuable and Figure 5 shows that the starting day influences the impact a lockdown can have on the maximum infection value.

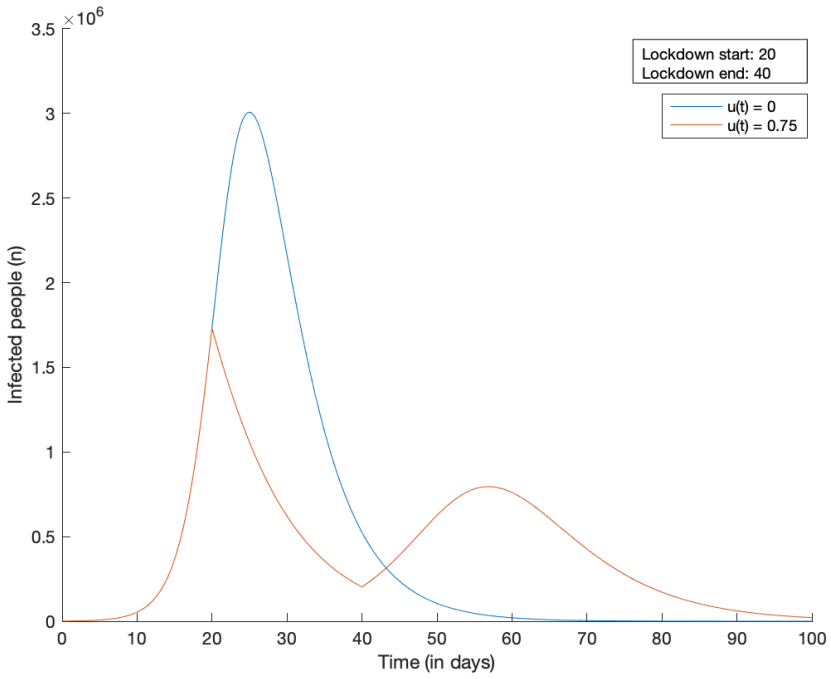

Figure 4: Infection values over time for $u(t) = 0$ and $u(t) = 0.75$ between $t_1 = 20$ and $t_2 = 40$.

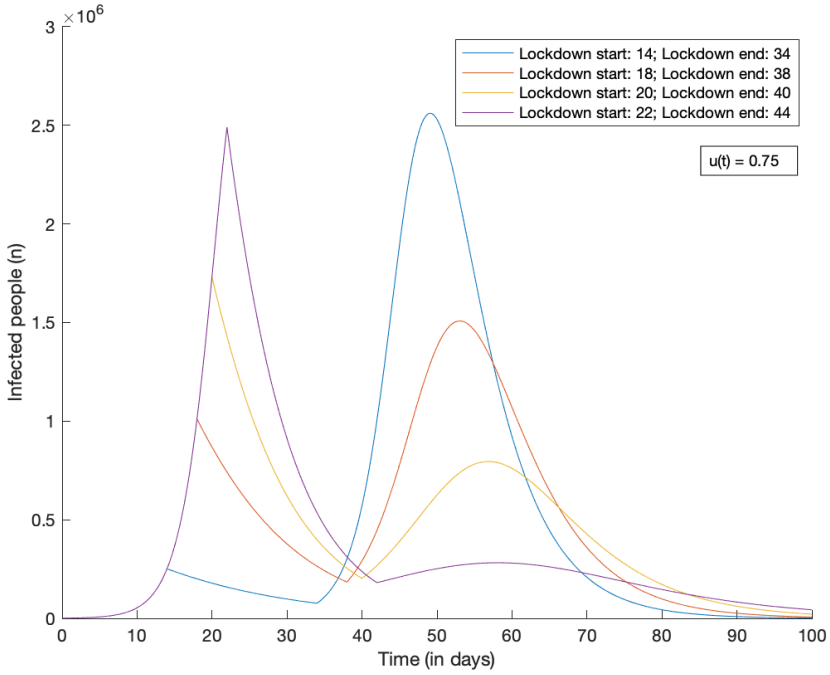

Figure 5: Infection values over time for starting days 14 (blue), 18 (red), 20 (yellow) and 22 (purple)

It was also valuable to analyze the impact of the parameters β and γ on the curves, since the values of these parameters can also strongly influence the maximum infection value, since these parameters are multiplied by the number of infections at a certain time (see equation 1). This was done by running the maximum infection values for all possible values for β and γ, using the SIR-model with the parameter values as in Table 1 and without lockdown (see Appendix E).

First, four different pairs of figures were plotted in which the limit of the maximum infection values were adjusted to see whether there exists fine structure (see Figure 6). Even though the four pairs did not show a fine structure, the plots do show that for values of $\beta \leq \gamma$ the maximum infection values are minimum. This means that when the transmission rate is equal or lower than the recovery rate, the maximum infection values are suppressed. This makes sense, since it is known that the reproduction number, which can be defined as the mean number of people that will be infected by one person and in the SIR-model is equal to $R_0 = \frac{\beta}{\gamma}$, will be equal or smaller than one, which means that for a lower transmission rate and higher recovery rate the spread of the disease diminishes and the maximum infection values will be suppressed, which indeed can be seen in the contour plot. For $\gamma \leq \beta$ the maximum infection values rapidly increase when γ and β increase. This makes sense as well, since the reproduction number will be equal to or greater than one for $\gamma \leq \beta$, which means that for a higher transmission rate and lower recovery rate the spread of the disease and the maximum infection values increase, which indeed can be seen in the contour plot. Additionally, the maximum infection values increase for an increase in $\beta \in [0.1, 1]$ and $\gamma \in [0, 0.5]$. This also makes sense, since the reproduction number will then increase in value, which is directly related with an increase in maximum infection values. Thus, the three-dimensional and contour plots show that different values for β and γ can have an impact on the maximum infection values: small changes between γ and β lead to

great differences in maximum infection values.

Next, three different three-dimensional and contour plots of the maximum infection values for all values of β and γ were plotted, using the SIR-model with the parameter values as in Table 1 to analyze the impact of implementing a lockdown on the maximum infection values and to see the differences between variations of lockdown strengths (see Appendix F). Figure 7 show plots for $u(t) = 0.35$, $u(t) = 0.55$, and $u(t) = 0.75$. All plots show a line of local minima at $\beta \approx 0.5$ and $\gamma \in [0, 0.3]$ whereas there is no line of local minima in the contour plot of the maximum infection values without lockdown (see first three-dimensional and contour plot in Figure 6). This means that implementing a lockdown leads to a line of local minima of the maximum infection values at $\beta \approx 0.5$ and $\beta \approx 0.5$ and $\gamma \in [0, 0.2]$. This means that suppressing the maximum infection values is not so much due to the lockdown strength, but rather to the fact that a lockdown is implemented. However, the plots also show that for a low value for the lockdown strength, the local minimum for high values of the maximum infection values is lower than for higher values of the lockdown strength. Thus, for an increase in the lockdown strength the local minimum rises, showing higher values of the maximum infection value. Additionally, an increase in the lockdown strength still leads to a minor change in an optimal value for the transmission rate β ($\beta \approx 0.4$ for $u(t) = 0.35$ versus $\beta \approx 054$ for $u(t) = 0.75$), which means that for higher given values of β the lockdown strength should also be higher, which makes sense since it is the transmission rate that increases but needs to be suppressed by the lockdown.

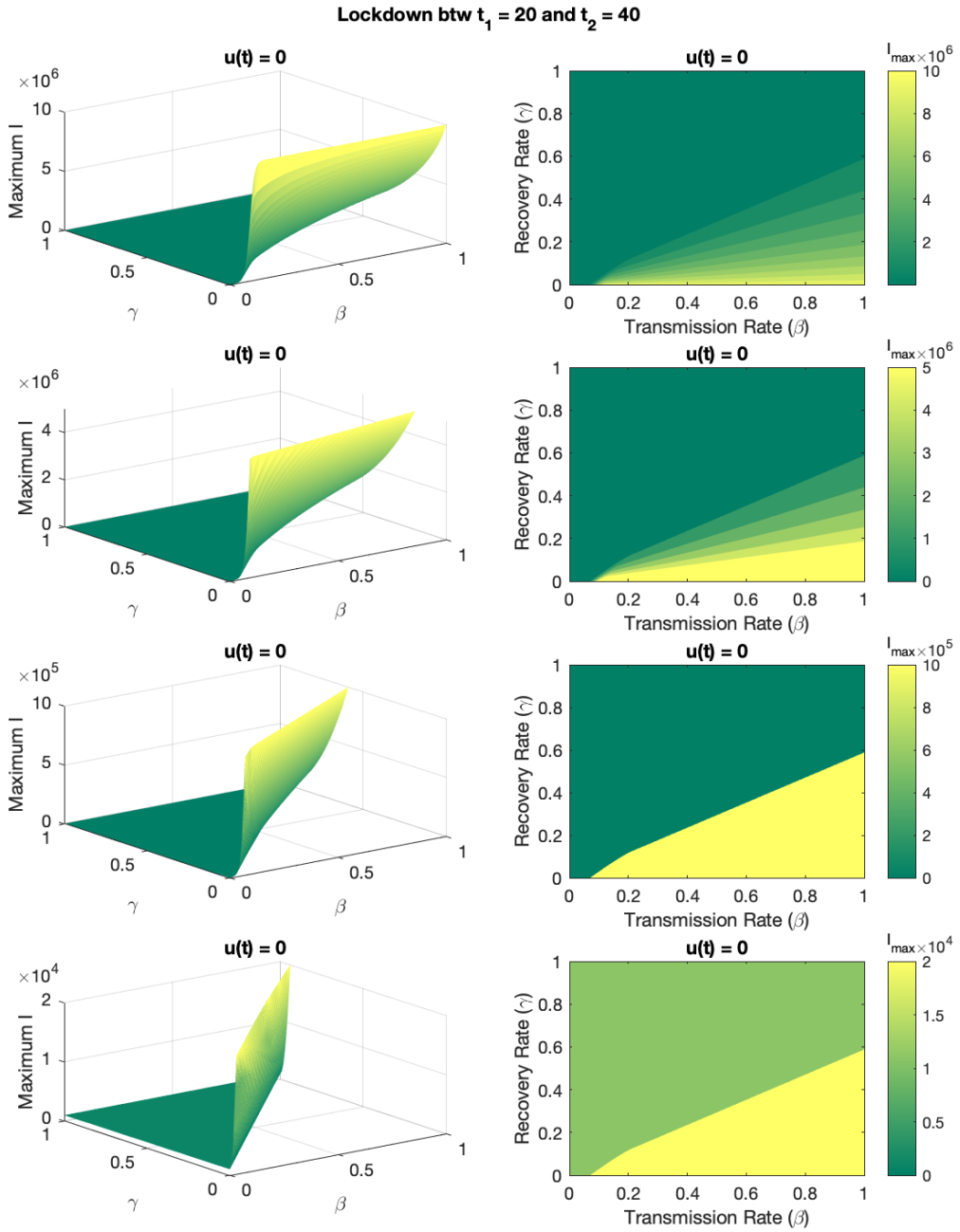

Figure 6: Three-dimensional (left) and contour (right) plots of maximum infection values for all values of β and γ for different limits of I_{max}

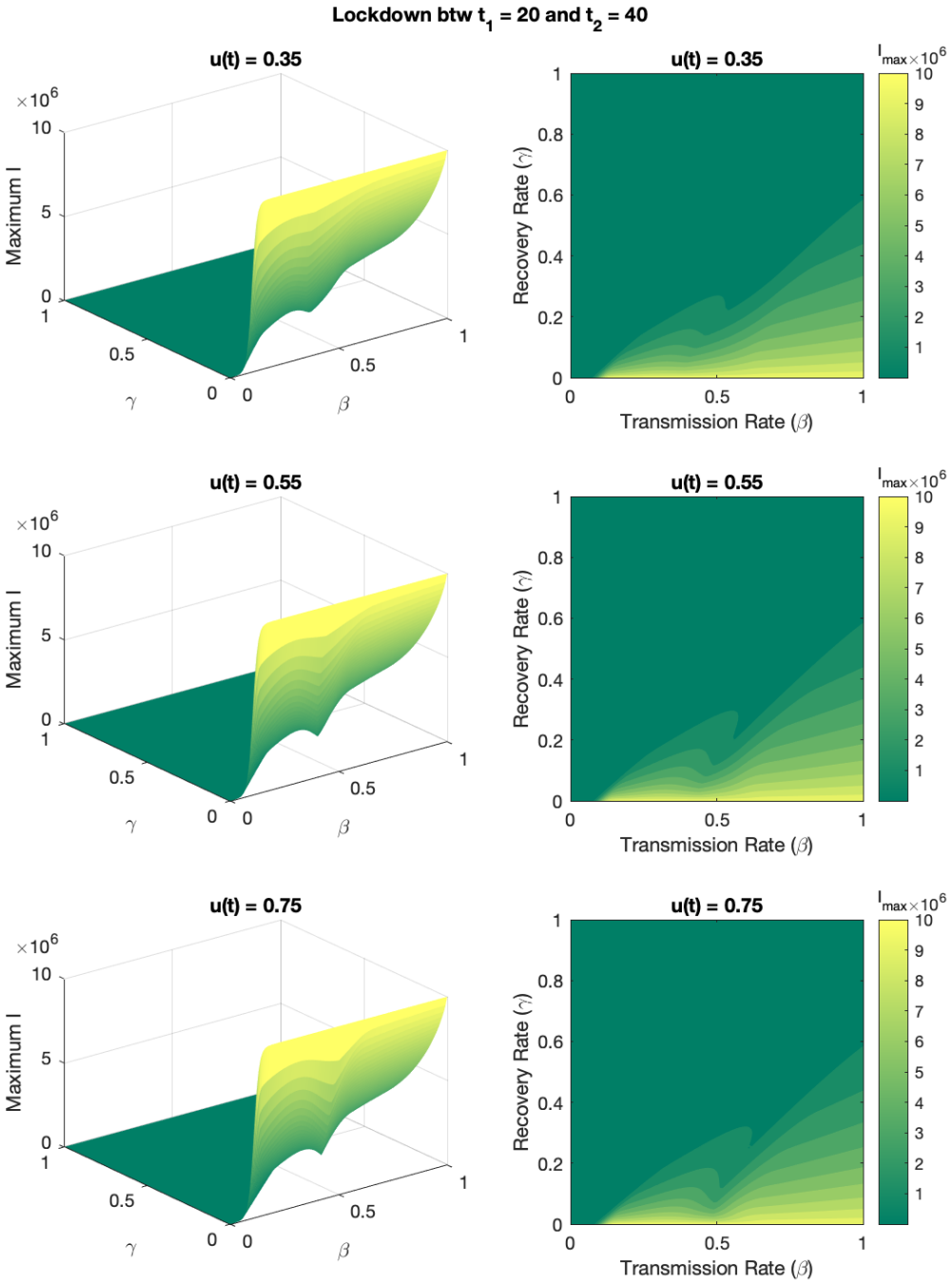

Figure 7: Three-dimensional (left) and contour (right) plots of maximum infection values for all values of β and γ for different lockdown strengths

3.1.3 The SIR-model with parameter values from COVID-19 data

In Section 3.1.2, we have gathered some notion on the influence of implementing a lockdown on the infection curve in the SIR-model and on the impact of different parameter values on the infection curve in the SIR-model. Now, it is possible to analyze the SIR-model using actual

COVID-19 data. Before running the SIR-model in MATLAB with the COVID-19 data, the values of the variables need to be determined in a way that could describe the period from 01-03-2020 until 01-06-2020 realistically. Therefore, it is necessary to choose the parameter values as they were in the period March 2020 until June 2020. Table 2 shows an overview of the parameter values used for the SIR-model; these values are based on research done on COVID-19 (see references in Table 2).

Table 2: List of parameters used in numerical simulations using the SIR-model

Parameter	***Value***	***Meaning***	***Reference***
N	17395687	Population size	[16]
t_{max}	92	Length of simulation (in days)	[31]
S(1)	17395675	Initial value of susceptible people	-
I(1)	12	Initial value of infected people	[3]
R(1)	0	Initial value of recovered people	[3]
γ	$\frac{1}{7}$	Recovery rate (per day)	[27]

The parameter value β varies per day and therefore, no fixed value of β for COVID-19 is available. However, it is possible to estimate β by the reproduction number (R_0), since R_0 is for the SIR-model defined as $R_0 = \frac{\beta}{\gamma}$ [8]. The same dataset as before ($owid - covid - data.csv$) also contained reproduction numbers for each day from 01-03-2020 until 01-06-2020. To determine a useful value for β, parametric fitting was used, which can be defined as the process in which a coefficient is found that can be fitted to the data. This was done with data of the reproduction numbers of the countries Italy, Spain, and France (see Appendices H, I, and J). The values of the reproduction numbers of the Netherlands were deliberately not used, since otherwise actual data from the Netherlands is used to simulate the infections over time in the Netherlands with the SIR-model to then compare this with the actual infections in the Netherlands. Thus, actual Dutch values would be used to compare with actual Dutch data, which would be incorrect to do. Figure 8 shows the reproduction numbers of these three countries over time together with the mean reproduction numbers per day (See Appendix G). The curve representing the mean reproduction numbers was determined by calculating the mean reproduction number of the countries Italy, France, and Spain per day. Therefore, Figure 8 shows reproduction numbers from 03-03-2020 until 01-06-2020, since this is a period in which all countries had data on reproduction numbers available allowing the infection curve representing the mean of the three countries to be calculated.

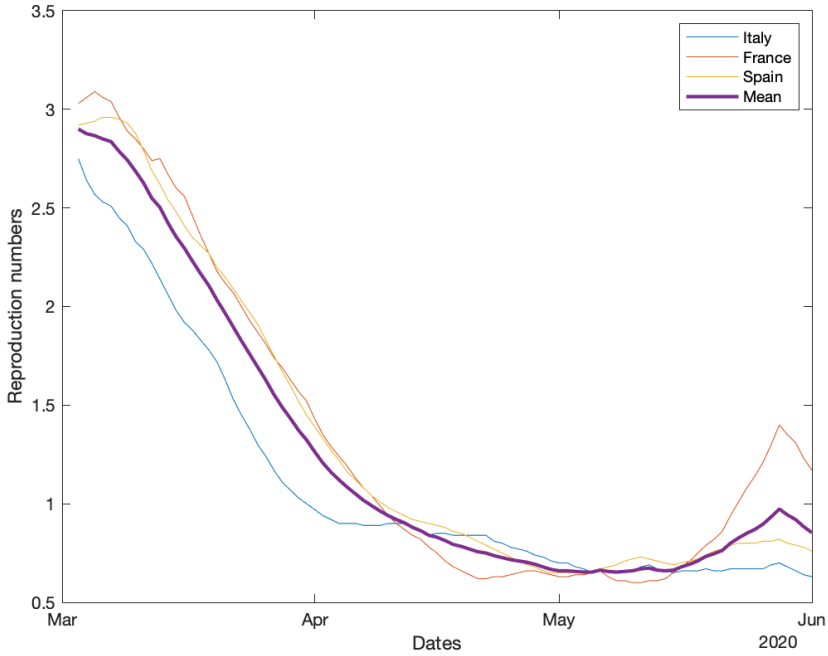

Figure 8: Reproduction numbers of Italy (blue), France (red), Spain (yellow), and mean of three countries (purple) over time

The estimated transmission rate β was determined by taking the mean of the estimated transmission rates of Italy, France, and Spain, β_{IT}, β_{FR}, and β_{SP}. The mean estimated transmission rates of Italy, France, and Spain were determined by multiplying the mean reproduction number of the respective country by the recovery rate, γ. The reproduction numbers of Italy, France, and Spain were determined by taking the mean of the daily reproduction numbers of each country respectively for the period before the respective country implemented its lockdown. The mean was calculated by using all days during the respective period of the country, since Figure 8 does not show outliers and more values provide a more accurate estimation for the value of the transmission rate of the respective country.

For Italy, there was data available on reproduction numbers from 24-02-2020; the lockdown in Italy was from 09-03-2020 until 04-05-2020 [3]. Thus, reproduction numbers for the period from 24-02-2020 until 09-03-2020 were used, i.e., before the lockdown was imposed. For France, there was data available on reproduction numbers from 01-03-2020; the lockdown in France was from 17-03-2020 until 11-05-2020 [3]. Thus, reproduction numbers for the period from 01-03-2020 until 17-03-2020 were used, i.e., before the lockdown was imposed. For Spain, there was data available on reproduction numbers from 03-03-2020; the lockdown in Spain was from 14-03-2020 until 21-06-2020 [3]. Thus, reproduction numbers for the period from 01-03-2020 until 17-03-2020 were used, i.e., before the lockdown was imposed. The mean reproduction numbers are equal to $R0_{IT} = 2.9313$, $R0_{FR} = 2.8856$, and $R0_{SP} = 2.8700$. The mean β of each country was calculated by multiplying the mean reproduction number of each country by $\gamma = \frac{1}{7}$. Therefore, the mean β values are equal to $\beta_{IT} = 2.9313 * 0.1428 = 0.4188$, $\beta_{FR} = 2.8856 * 0.1428 = 0.4122$, and $\beta_{SP} = 2.8700 * 0.1428 = 0.4100$ for Italy, France, and Spain respectively. The estimated value for β, which was taken to be the mean of β_{IT}, β_{FR},

β_{SP}, is equal to $(0.4188 + 0.4122 + 0.4100)/3 = 0.4137$. Therefore, we take $\beta = 0.4137$ as the independent estimate for the transmission rate.

During the lockdown, the reproduction number for the SIR-model with lockdown is equal to $R_0 = \frac{\beta(1-u)}{\gamma}$, and since R_0, γ, and now also β are known, it is possible to estimate the value c for $u(t) = c$. As mentioned in Section 3.1.1, the lockdown strength was chosen in its simplest form, namely as a constant function. The lockdown strength $u(t) = c$ was determined by taking the mean of the estimated lockdown strengths of Italy, France, and Spain of each day during the lockdown in the respective country, since using more values provides a more accurate estimation for the value of the lockdown strength of the respective country. The mean estimated lockdown strength per country was calculated as $u(t)_{country} = 1 - \frac{R_{0country}(t)\gamma}{\beta}$, in which the reproduction numbers for the periods 09-03-2020 until 04-05-2020, 17-03-2020 until 11-05-2020, and 14-03-2020 until 21-06-2020 were used for Italy, France, and Spain respectively, since these were the periods there was a lockdown in the respective country. It is for the same reason as for calculating the independent estimate for the transmission rate that we deliberately did not use reproduction numbers of the Netherlands, but those of Italy, France, and Spain again (see Appendix H, I, and J. Figure 9 shows the values of the lockdown strength of the three countries over time together with the mean curve for the lockdown strength (see Appendix G). The latter was determined by taking the daily mean of the calculated daily lockdown strength values of Italy, France, and Spain. The estimated lockdown strength values for each country were $u_{IT} = 0.5958$, $u_{FR} = 0.6185$, $u_{SP} = 0.6211$ for Italy, France, and Spain respectively. The estimated value for u, which was taken to be the mean of u_{IT}, u_{FR}, u_{SP} is then equal to $(0.5958 + 0.6185 + 0.6211)/3 = 0.6118$. Therefore, we take $u = 0.6118$ as the independent estimate for the lockdown strength.

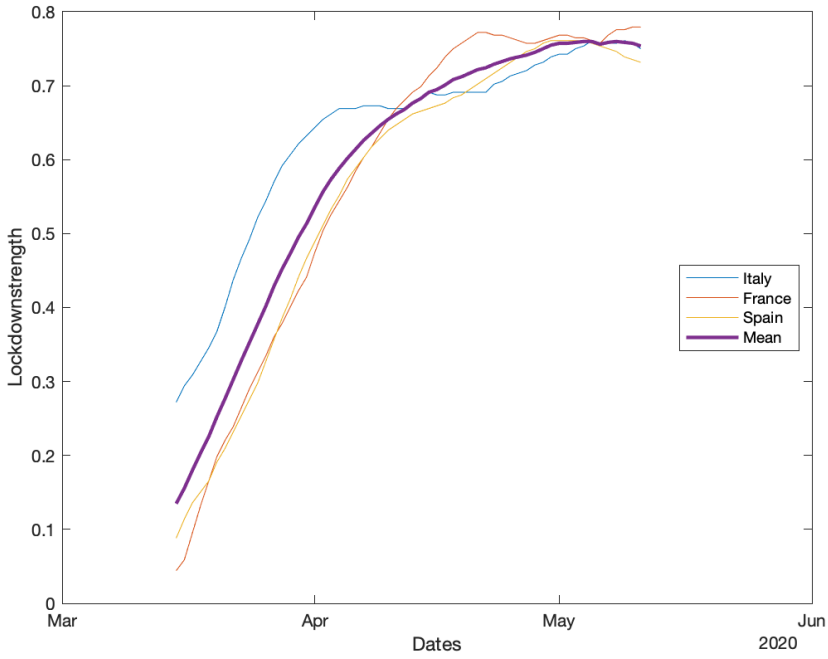

Figure 9: Lockdown strength of Italy (blue), France (red), Spain (yellow), and mean of three countries (purple) over time

Now, as all values for the parameters are known, it is possible to plot the SIR-model that should describe the infections as they were in the Netherlands for the period 01-03-2020 until 01-06-2020. Figure 10 shows the infection curve with and without lockdown with the parameter values as in Table 2, the transmission rate equal to $\beta = 0.4137$, and the lockdown strength equal to $u(t) = 0.6118$ (see Appendix K).

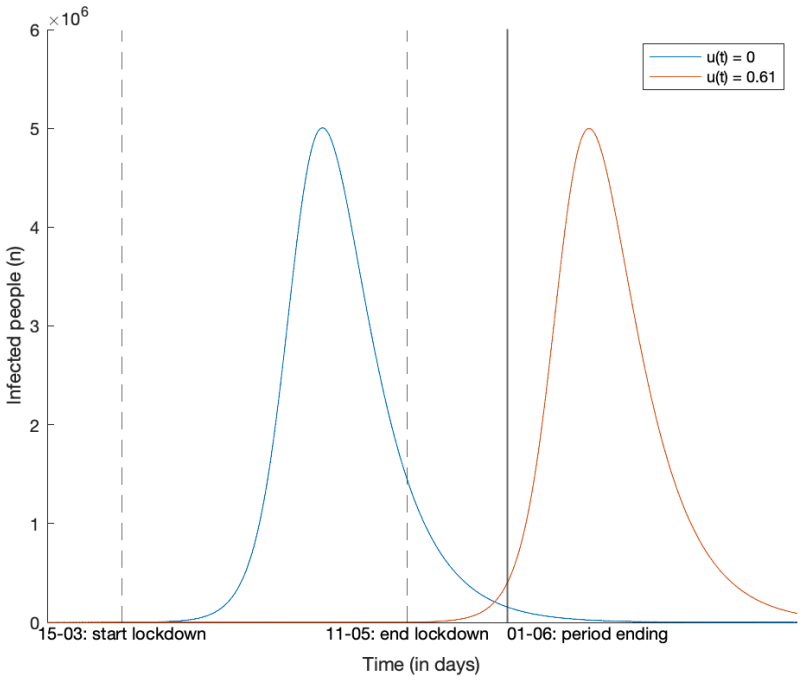

Figure 10: SIR-model over time with and without lockdown according to COVID-19 data

As mentioned in Section 3.1.1, the effectivity of the lockdown for this SIR-model can be determined by comparing the maximum infection value without lockdown with the maximum infection value with lockdown. However, before determining this effectivity, it is crucial to analyze whether the SIR-model with lockdown and parameter values as in Table 2 adequately describes the period prevalence of COVID-19 in the Netherlands (see Figure 1). This can be done by comparing the infection curves in the SIR-model with the period prevalence curve. We can also apply our priorknowledge of the theoretical SIR-model to say something about the current parameter fitting.

When comparing the infection curve with lockdown with the period prevalence curve, we observe that they are not similar since the maximum infection peak in the SIR-model is much higher, 4,998,200 in the SIR-model with lockdown versus 11,047 in the period prevalence curve (see Figure 1). However, it is because of the population size $N = 17,395,687$ that the infection peak can be equal to $4,998,200$. This means that the population size chosen in the SIR-model is too big, which could indeed be the case, since COVID-19 hardly occurred in people up to and including 60 years of age and was mostly relevant for people older than 60 years [20].

Additionally, the infection peak in the SIR-model is much later, 17-06-2020 in the SIR-model with lockdown versus 18-04-2020 in the period prevalence curve. As mentioned in Section 3.1.1, $I(1)$ is very small, and because of that, more time is needed for the infection curve to form the

whole curve, which explains why both infection curves in Figure 4 are moved to the right and have their infection peak at a much later moment than there was in real life. Thus, even though the infection curve of the SIR-model does not necessarily look much like the period prevalence curve, the major differences were to be expected.

When considering the effectivity of the lockdown in the SIR-model, the effectivity of the lockdown is determined by comparing the maximum infection value without lockdown with the maximum infection value with lockdown. This is done for the period the infection curves with and without lockdown is fully visible instead for the period 01-03-2020 until 01-06-2020, since otherwise the maximum infection value for the infection curve with lockdown would be equal to 0, which does not fairly represent the true effect of the lockdown in the SIR-model. The effectivity of the lockdown is determined by the following quantity:

$$E_{SIR} := \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\%. \tag{4}$$

For the SIR-model with lockdown and the parameter values as in Table 2, we have $I_{max} = 4,864,200$ infections, and $I_{max+u(t)} = 4,864,200$ infections (see Figure 10). This means that the effectivity of the lockdown was equal to

$$E_{SIR} = \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\% = \left(1 - \frac{4,998,200}{5,004,100}\right) * 100\% = 0.12\%. \tag{5}$$

Thus, the effect of the lockdown for the SIR-model with parameter values as in Table 2 is negligible, since a change of 0.12% in maximum infection values will not have a clear impact on pressure on healthcare systems due to this maximum infection value. However, this apparent negligible effect of the lockdown on the maximum infection value can be explained: as mentioned in Section 3.1.1, the ratio between the initial value of infected people, $I(1)$, and the population size should be around $\frac{1}{10000}$ for the value of β and $u(t)$ to have an observable effect. However, the ratio for the SIR-model with lockdown and the parameters as in Table 2 is equal to $\frac{I(1)}{N} = \frac{12}{17395687} = 6.898 * 10^{-7}$, which is too small. Nevertheless, this makes sense, since the population size is in all likelihood too large. Also, the initial value of infected people is equal to 12, but these were infections reported in hospital [3]. It could very well be that the true initial value of infected people was much higher (and not reported at that time). Also, Figure 11 show a three-dimensional and contour plot of maximum infection values for all values of β and γ according to COVID-19 data. From the figure and Section 3.1.2 we know that the estimated value for β, and therefore also the estimated value for $u(t)$, now is somewhat low and could have greater impact on the maximum infection value when it would have been greater ($\beta \approx 0.5$) (see Appendix L). Therefore, the infection curve with lockdown in Figure 10 can be considered a representation of the impact of the ratio between the population size and initial infected people on the infection curve, rather than the impact of implementing a lockdown on the infection curve.

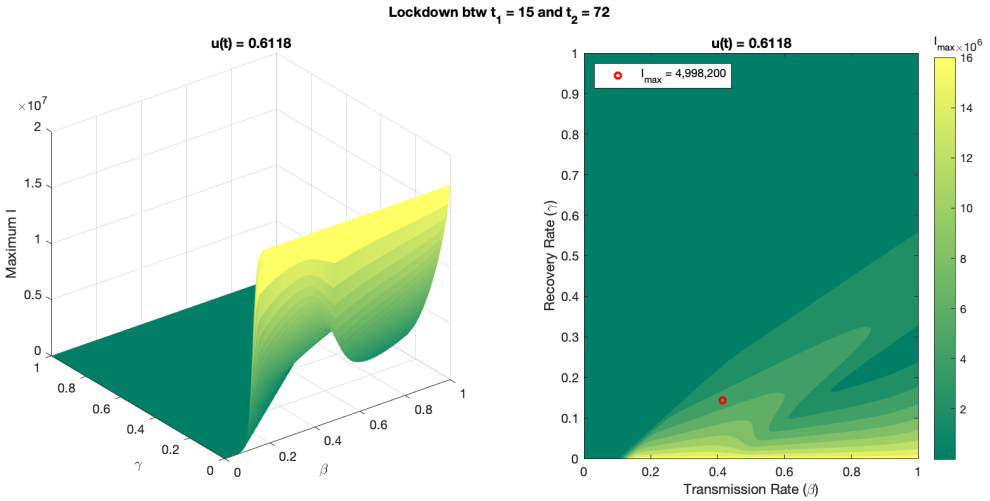

Figure 11: Three-dimensional (left) and contour (right) plots of maximum infection values for all values of β and γ according to COVID-19 data

In consideration of the above, it is valuable to apply some more parameter fitting, to see whether we get a more adequate SIR-model and a more visible effect of the lockdown. For the infection curve in the SIR-model, the population size was adjusted to the population size of people aged 61 or older, which in March 2020 was equal to approximately 3.4 million people [15]. Also, the value for $I(1)$ was changed to $I(1) = 1200$ to obtain a proper ratio of $I(1)$ and N and to make sure the infection peak is during the period of the lockdown. Additionally, to obtain a more realistic analysis of the effect of the lockdown, instead of using the mean of $u(t)$, which was equal to $u(t) = 0.6118$, $u(t)$ was used as a time-dependent function in which the values for $u(t)$ change per day and are according to the mean lockdown curve as in Figure 9.

To observe the impact of the change in parameter values separately, the population size and initial infection value were first changed, since visualisation of the impact of implementing a lockdown is dependent on a proper ratio between the population size and initial infection value. Therefore, the population size and initial infection value were both changed at the same time. Figure 12 shows the SIR-model over time with and without lockdown with new parameter values for the population size and initial infection value. The figure shows that because of the decrease in population size the maximum infection values for both the infection curve with and without lockdown has decreased greatly. Also, both infection curves have moved to the left. The impact of implementing a lockdown now is more evident (see Appendix M).

Figure 13 shows the SIR-model over time with and without lockdown with both new parameter values for population size and initial infection value as well as a time-dependent lockdown strength. Changing the lockdown strength to a time-dependent function upon changing the parameter values of the population size and intial infection value was done explicitly, since a fair visualisation of the impact of implementing a lockdown is dependent on a proper ratio between the population size and the initial infection value, which would not have been the case using the population size and initial infection value as in Table 2. The figure shows clearly the impact of implementing a time-dependent, and therefore more realistic, lockdown (see Appendix M).

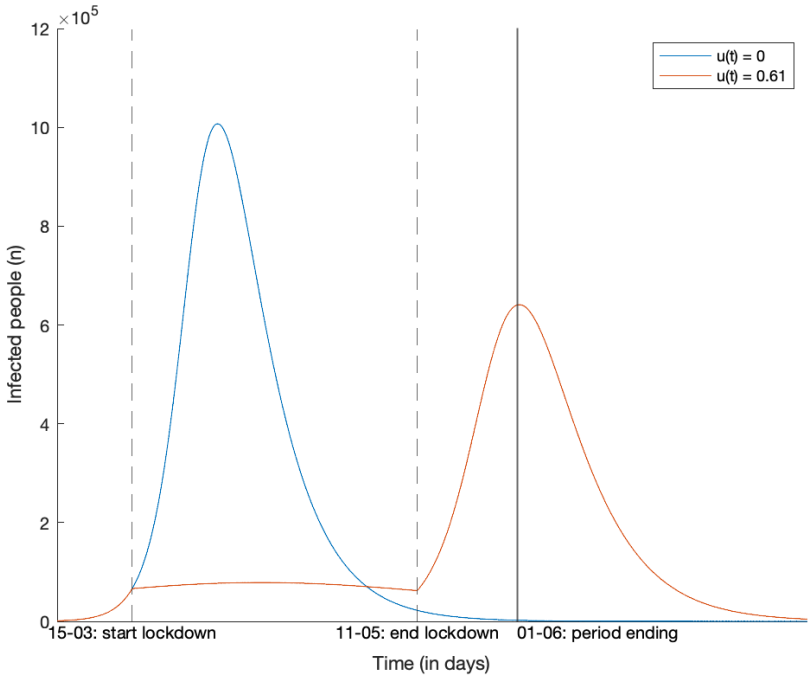

Figure 12: SIR-model over time with and without lockdown with new parameter values for the population size and initial infection value

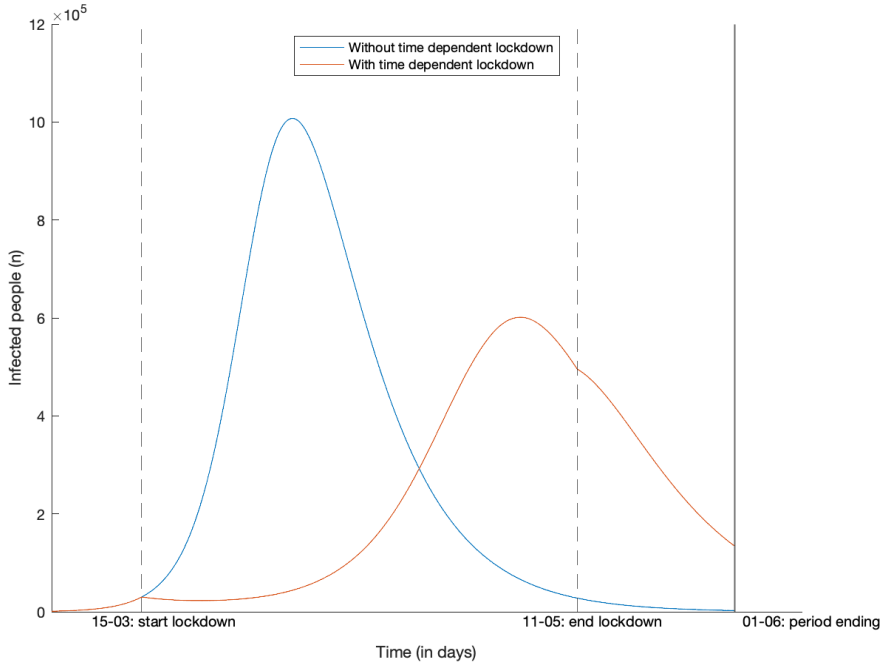

Figure 13: SIR-model over time with and without lockdown with new parameter values for population size and initial infection value and a time-dependent lockdown strength

Applying the new parameters to the SIR-model has thus proven to provide a more realistic infection curve with and without lockdown, since it showed more resemblance with the period prevalence curve: the infection curve without lockdown is exactly between the lockdown period

because of our new value for $I(1)$; and implementing a lockdown shows clearly an effect because of applying $u(t)$ as a time-dependent function. For the SIR-model with lockdown and the new parameter values for $I(1)$ and $u(t)$ respectively as in Figure 13, we have $I_{max} = 582,480$ and $I_{max+u(t)} = 978,470$ (see Figure 13). This means that the effectivity of the lockdown was equal to

$$E_{SIR} = \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\% = \left(1 - \frac{582,480}{978,470}\right) * 100\% = 40.5\%. \tag{6}$$

Thus, the effect of the lockdown for the SIR-model with new parameter values for $I(1)$ and $u(t)$ can be considered substantial, since a change of 59.7% can be considered impactful with respect to pressure on healthcare systems due to the maximum infection value. Assuming that the infection curve with lockdown in this SIR-model simulates the period prevalence of COVID-19 in the Netherlands sufficiently, it can be concluded that implementing the lockdown in the Netherlands at that time was effective enough to control the situation at that time.

3.2 The SEIRS-model

3.2.1 Theoretical background

The above demonstrates that the lockdown implemented in the Netherlands on 11-03-2020 could presumably be considered as an effective measure to control the spread of the virus across the country. However, the theoretical model used in the previous section was the SIR-model. To simulate reality even more adequately, it is desirable to apply a theoretical model that describes reality more accurately, since it reduces the number of assumptions of the SIR-model relevant for the period from March to June 2020. The relevant theoretical model is the so-called SEIRS-model (**s**usceptible, **e**xposed, **i**nfected, **r**ecovered, and **s**usceptible model), and takes away all assumptions mentioned in Section 3.1.1, and takes away all assumptions mentioned in Section 3.1.1, except for the assumption that the number of vaccinations remains constant over time [8]. However, this does not cause a problem, since there were no vaccinations yet at that period of time. The SEIRS-model expands on the SIR-model by adding an incubation time and the possibility to become susceptible again after recovery. Just as in the SIR-model, it is possible to add the constant function $u(t)$ to the model representing the lockdown. The SEIRS-model with lockdown can be expressed by the following system of ordinary differential equations:

$$\begin{cases} \frac{dS}{dt} = \mu N - \mu S - \frac{(1-u)\beta IS}{N} + \omega R, \\ \frac{dE}{dt} = \frac{(1-u)\beta IS}{N} - \sigma E - \mu E, \\ \frac{dI}{dt} = \sigma E - \gamma I - (\mu + \alpha) I, \\ \frac{dR}{dt} = \gamma I - \mu R - \omega R, \end{cases} \tag{7}$$

with $u(t)$ as in equation 3. Here, $\frac{dS}{dt}$, $\frac{dE}{dt}$, $\frac{dI}{dt}$, and $\frac{dR}{dt}$ represent the rate of change in the number of people that are in the S, E, I or R category respectively with respect to time. The parameter μ represents the birth and death rate, β the transmission rate, ω the rate of loss of immunity, σ the incubation period, α the rate of death through disease, and γ the recovery rate. N represents the population size. Figure 14 shows a schematic overview of the SEIRS-model with all parameters

mentioned above, as well as the cost function $u(t)$.

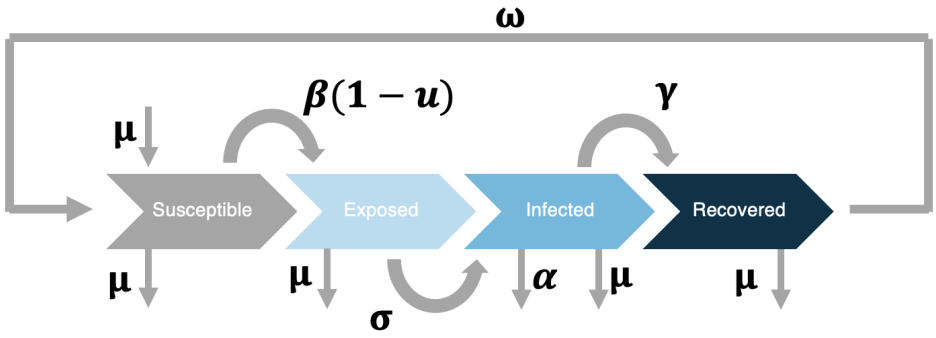

Figure 14: Schematic overview of SEIRS-model

3.2.2 The SEIRS-model with arbitrary parameter values

Before applying the actual data, it is again valuable to analyze the influence of implementing a lockdown function into the SEIRS-model to gather some notion on how a lockdown can influence the infection curve. This was done by running the SEIRS-model in MATLAB with parameter values arbitrarily chosen. Table 3 shows an overview of the parameters used for the SEIRS-model. The lockdown was set on $u(t) = 0.75$ for $t_1 = 80$ to $t_2 = 100$ (see Appendix N).

Table 3: List of arbitrary parameters used in numerical simulations for SEIRS-model

Parameter	***Value***	***Meaning***
N	10000000	Population size
t_{max}	200	Length of simulation
S(1)	9999000	Initial value of susceptible people
E(1)	1000	Initial value of exposed people
I(1)	0	Initial value of infected people
R(1)	0	Initial value of recovered people
γ	$\frac{1}{14}$	Recovery rate
β	0.6	Transmission rate
μ	$\frac{1}{76*365}$	Birth and death ratio
ω	$\frac{1}{365}$	Rate of loss of immunity
σ	$\frac{1}{7}$	Latency period
α	0.2	Infection-induced death ratio

Figure 15 shows two infection curves visualising the number of infections over time with and without lockdown. The curves clearly show the impact of implementing a lockdown, since the maximum infection value decreases when a lockdown is implemented in the model. This makes sense, since the fraction $\frac{\beta IS}{N}$ is multiplied by $1 - 0.75$ and which leads to a decrease in the maximum value of this fraction and thus the maximum infection value. Also, whereas the initial value of infected people, $I(1)$, influences the position of the infection curve in the SIR-model, the initial value of exposed people, $E(1)$ influences the position of the infection curve in the SEIRS-model. The lower the value of $E(1)$, the more the infection curve is positioned to the

right, since more time is needed for the infection values to increase. Additionally, multiple simulations of the SEIRS-model with different arbitrary parameter values for the population size and the initial value of exposed people showed that a ratio of $\frac{1}{10000}$ of the initial value of exposed people relative to the population size is needed to see the impact of β (and $u(t)$) on the infection curve. Lastly, multiple simulations of the SEIRS-model when changing arbitrarily parameter values for β, γ, μ, ω, σ, and α separately from each other show the following: an increase in β decreases the maximum infection value, whereas the infection curve to starts to increase earlier; an increase in γ decreases the maximum infection value; an increase in μ causes the limit of the infection curve to move towards a constant value other than zero; an increase in ω causes the limit of the infection curve to also move towards a constant value other than zero; an increase in α causes the infection curve to be spread over a longer period of time and the maximum infection value to increase in value; and an increase in σ causes the infection curve to be spread over a shorter period of time and the maximum infection value to decrease.

Figure 16 shows infection curves for different starting days of a lockdown given a duration of 20 days (see Appendix O). The plots show that for different starting days the lockdown has a different influence on the infection values. For an early starting day (blue), there first will be a lower peak and then a higher peak. This makes sense, since the lockdown is implemented during the increase in infection values, which means that the lockdown has less influence during the peak of the lockdown; for a starting day around the peak of the infection curve without lockdown (red), there will be two peaks at approximately the same height. This also makes sense, since the lockdown is implemented during the peak in infection values, which means that the lockdown has optimal influence on the maximum infection values. For a starting day around the decrease in infection values of the infection curve without lockdown (yellow), there will be a higher peak and then a lower peak. This again makes sense, since the lockdown is implemented after the peak in infection values is reached, which means that the lockdown has less influence during the peak of the lockdown. Thus, the same can be concluded as in Section 3.1.2 for the SIR-model: Figure 15 shows that implementing a lockdown can be valuable and Figure 16 shows that the starting day influences the impact a lockdown can have on the maximum infection value. A main difference between the SIR- and SEIRS-model is that the infection curve in the SEIRS-model is spread over a longer period of time (200 days instead of 100) and starts to increase later, which makes sense, since it now takes longer for a population to become infected, because a person first becomes exposed. Since the infection curve start to increase later, it is easier for the SEIRS-model to have a lockdown that is implemented too early (purple). In this case, the lockdown has no effect on the maximum infection value, and implementing a lockdown will only cause the infection curve to increase even later than it already does. Another main difference is that the maximum infection value are considerably lower in the SEIRS-model.

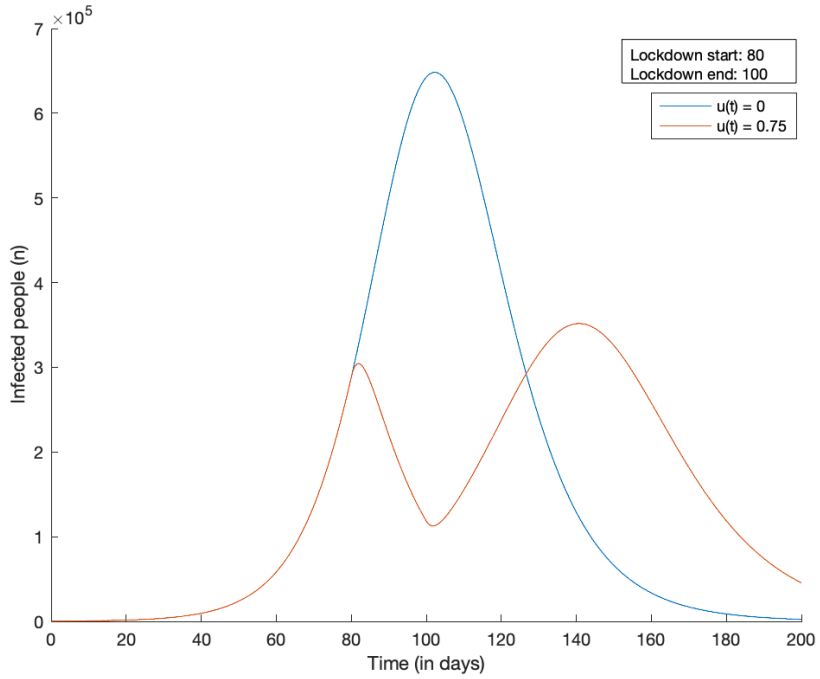

Figure 15: Infection values over time for u(t) = 0 and u(t) = 0.75 between $t_1 = 80$ and $t_2 = 100$

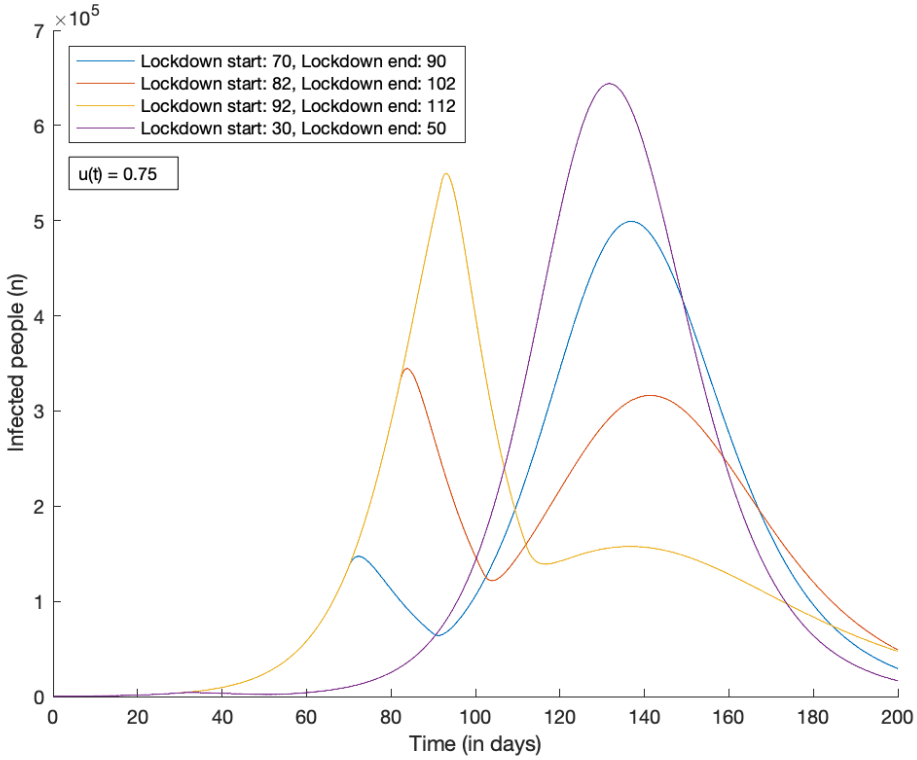

Figure 16: Infection values over time for starting days 70 (blue), 82 (red), 92 (yellow), and 30 (purple)

It was also valuable to analyze the influence the impact of β and γ on the curves, since the

values of these parameters can also strongly influence the maximum infection value, since these parameters are multiplied by the number of infections at a certain time (see equation 7). This was done by running the maximum infection values for all possible values for β and γ, using the SEIRS-model with the parameters as in Table 3 and without lockdown.

Firstly, four different pairs of figures were plotted in which the limit of the maximum infection values were adjusted to see whether there exists fine structure (see Appendix P). Even though the four pairs did not show a fine structure, the plots do show that for values of $\beta \leq \gamma$ the maximum infection values are minimum. This means that when the transmission rate is equal or lower than the recpovery rate, the maximum infection values are suppressed. This makes sense, since it is known that the reproduction number, which can be defined as the mean number of people that will be infected by one person and in the SEIRS-model is equal to $R_0 = \frac{\sigma}{\sigma+\mu} * \frac{\beta(1-u)}{\alpha+\gamma+\mu}$, will be equal or smaller, which means that for a lower transmission rate and higher recovery rate the spread of the disease diminishes and the maximum infection values will be suppressed, which indeed can be seen in the contour plot. For $\gamma \leq \beta$ the maximum infection values rapidly increase when γ and β increase. This makes sense as well, since the reproduction number will be equal to or greater than one for $\gamma \leq \beta$, which means that for a higher transmission rate and lower recovery rate the spread of the disease and the maximum infection values increase, which indeed can be seen in the contour plot. Additionally, the maximum infection values increase for an increase in $\beta \in [0.25, 1]$ and $\gamma \in [0, 0.4]$. This also makes sense, since the reproduction number will then increase in value, which is directly related to an increase in maximum infection values. Thus, the three-dimensional and contour plots show that different values for β and γ can have an impact on the maximum infection values: small changes between γ and β lead to substantial differences in maximum infection values.

Next, three different three-dimensional and contour plots of the maximum infection values for all values of β and γ were plotted, using the SIR-model with the parameter values as in Table 3 to analyze the impact of implementing a lockdown on the maximum infection values and to see the differences beween variations of lockdown strengths (see Appendix Q). Figure 18 show plots for for $u(t) = 0.35$, $u(t) = 0.55$, and $u(t) = 0.75$. All plots show that a line of local minima arises for values from $\gamma \approx 0$ to $\gamma \approx 0.3$ and from $\beta \approx 0.3$ to $\beta \approx 1$ whereas there is no line of local minima in the contour plot of the maximum infection values without lockdown (see first three-dimensional and contour plot in Figure 17). This means that implementing a lockdown leads to a line of local minima of the maximum infection values from $\gamma \approx 0$ to $\gamma \approx 0.3$ and from $\beta \approx 0.3$ to $\beta \approx 1$. The plots also show that for a low value for the lockdown strength, the local minimum for high values of $\beta \in [0.3, 1]$ and $\gamma \in [0, 0.3]$ the maximum infection values are lower than for higher values of the lockdown strength. Thus, for an increase in the lockdown strength the maximum infection values decrease faster, which makes sense since the lockdown strength has a direct impact on the maximum infection values.

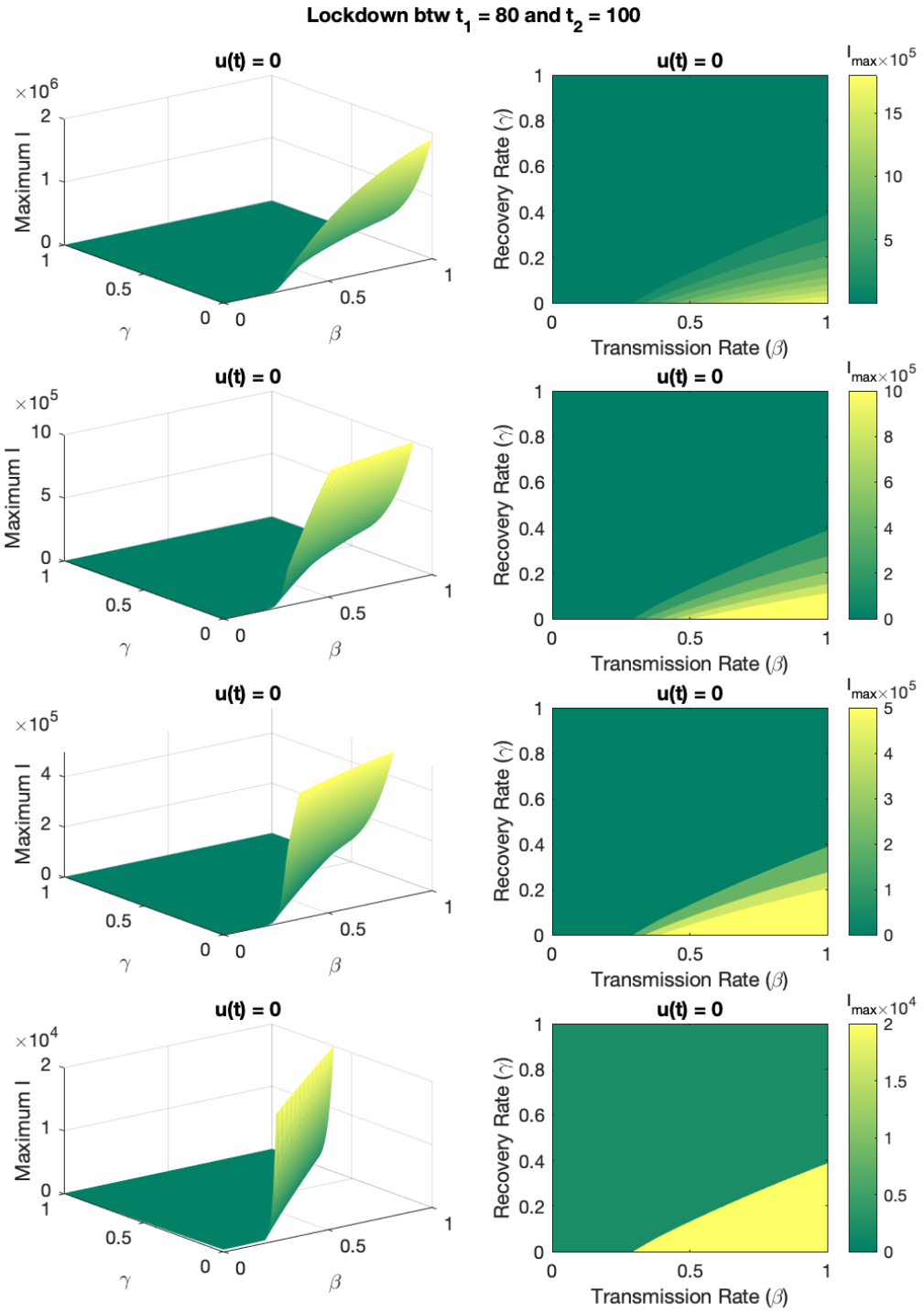

Figure 17: Three-dimensional (left) and contour (right) plots of maximum infection values for all values of β and γ for different limits of I_{max}

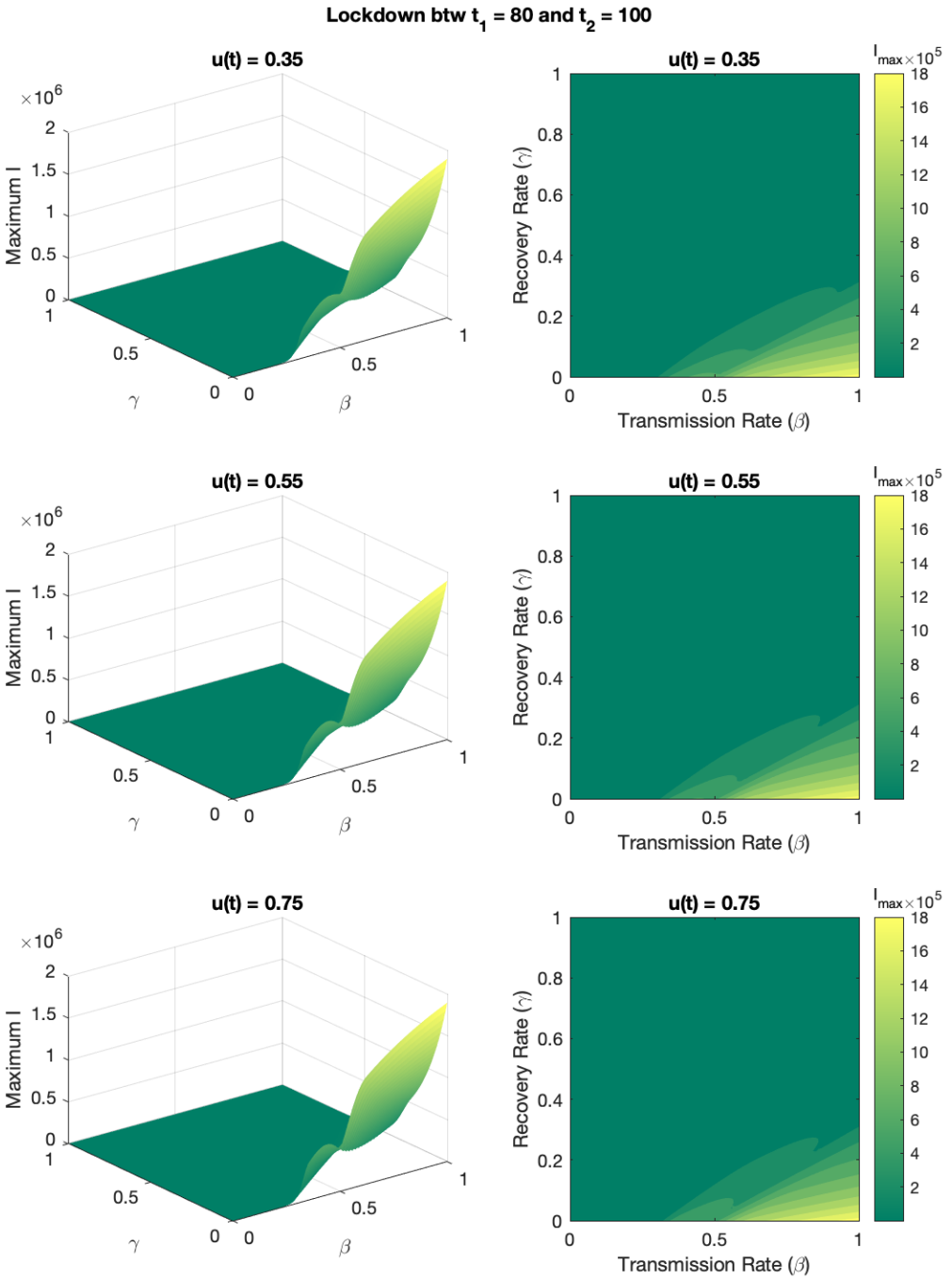

Figure 18: Three-dimensional (left) and contour (right) plots of maximum infection values for all values of β and γ for different lockdown strengths

3.2.3 The SEIRS-model with parameter values from COVID-19 data

In Section 3.2.2, we have gathered some notion on the influence of implementing a lockdown on the infection curve in the SEIRS-model and on the impact of different parameter values on the infection curve in the SEIRS-model. Now, it is possible to analyze the SEIRS-model using actual

COVID-19 data. Before running the SEIRS-model in MATLAB with the COVID-19 data, the values of the variables need again to be determined in a way that could describe the period from 01-03-2020 until 01-06-2020 realistically. Therefore, it is necessary to choose the parameter values as they were in the period from March 2020 until June 2020. Table 4 shows an overview of the parameter values used for the SEIRS-model; these values are based on research done on COVID-19 (see references in Table 2).

Table 4: List of parameters used in numerical simulations for SEIRS-model

Parameter	***Value***	***Meaning***	***Reference***
N	17395687	Population size	[16]
t_{max}	92	Length of simulation (in days)	[31]
S(1)	17395675	Initial value of susceptible people	-
E(1)	12	Initial value of people in incubation period	[3]
I(1)	0	Initial value of infected people	[3]
R(1)	0	Initial value of recovered people	[3]
γ	$\frac{1}{7}$	Recovery rate (per day)	[27]
μ	$\frac{1}{76*365}$	Birth \death rate (per day)	[8]
ω	$\frac{1}{365}$	Rate of loss of immunity (per day)	[8]
σ	$\frac{1}{7}$	Latency period (in days)	[8]
α	0	Infection-induced death ratio (per day)	[8]

The parameter value β varies per day and therefore, no fixed values of β for COVID-19 are available. However, it is again possible to estimate β by the reproduction number (R_0). For the SEIRS-model we have that $R_0 = \frac{\sigma}{\sigma+\mu} * \frac{\beta}{\alpha+\gamma+\mu}$ [8]. Parametric fitting was done again using the dataset $owid-covid-data.csv$ containing reproduction number of Italy, Spain, and France. As for the same reason mentioned in Section 3.1.3, reproduction numbers of the Netherlands were deliberately not used. Figure 8 shows the reproduction numbers of these three countries over time and is the same as used in Section 3.1.3. Figure 8 shows the reproduction numbers of these three countries over time (see Appendix G).

The mean values of β of each country were calculated in exactly the same manner as in Section 3.1.3 except for the fact that R_0 was changed from $R_0 = \frac{\beta}{\gamma}$ into $R_0 = \frac{\sigma}{\sigma+\mu} * \frac{\beta}{\alpha+\gamma+\mu}$. The mean values of β of each country values are equal to $\beta_{IT} = 0.4190$, $\beta_{FR} = 0.4124$, and $\beta_{SP} = 0.4102$ for Italy, France, and Spain respectively. The estimated value for β, which was again taken to be the mean of β_{IT}, β_{FR}, and β_{SP}, is equal to $(0.4190+0.4124+0.4102)/3 = 0.4139$. Therefore, we take $\beta = W$ as the independent estimate for the transmission rate (see Appendix R).

During the lockdown, the reproduction number for the SEIRS-model with lockdown is equal to $R_0 = \frac{\sigma}{\sigma+\mu} * \frac{\beta(1-u)}{\alpha+\gamma+\mu}$, and since R_0, γ, μ, σ, α and now also β are known, it is possible to estimate the value c for $u(t) = c$ using the fact that $u(t) = 1 - \frac{R_0(\sigma+\mu)(\alpha+\gamma+\mu)}{\sigma\beta}$. The value for $u(t)$ was calculated in the same manner as in Section 3.1.3, using the reproduction numbers of Italy, France, and Spain again. The estimated $u(t)$ values for each country were $u_{IT} = 0.5959$, $u_{FR} = 0.6186$, $u_{SP} = 0.6212$ for Italy, France, and Spain respectively. The estimated value for $u(t)$, which was taken to be the mean of u_{IT}, u_{FR}, u_{SP}, is equal to $(0.5959+0.6186+0.6212)/3 =$

0.6119. Therefore, we take $u = 0.6119$ as the independent estimate for the lockdown strength (see Appendix R).

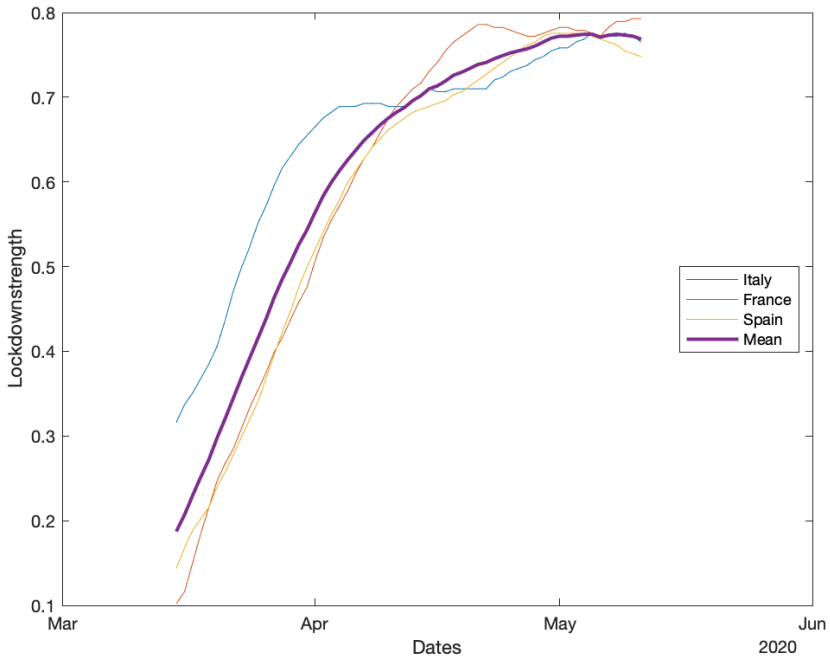

Figure 19: Lockdown strength of Italy (blue), France (red), Spain (yellow), and mean of three countries (purple) over time

Now, as all values for the parameters are known, it is possible to plot the theoretical model that should describe the infections as they were in the Netherlands for the period 01-03-2020 until 01-06-2020. Figure 20 shows the infection curve with and without lockdown with the parameter values as in Table 4, the transmission rate equal to $\beta = 0.4139$, and the lockdown strength equal to $u(t) = 0.6119$ (see Appendix S).

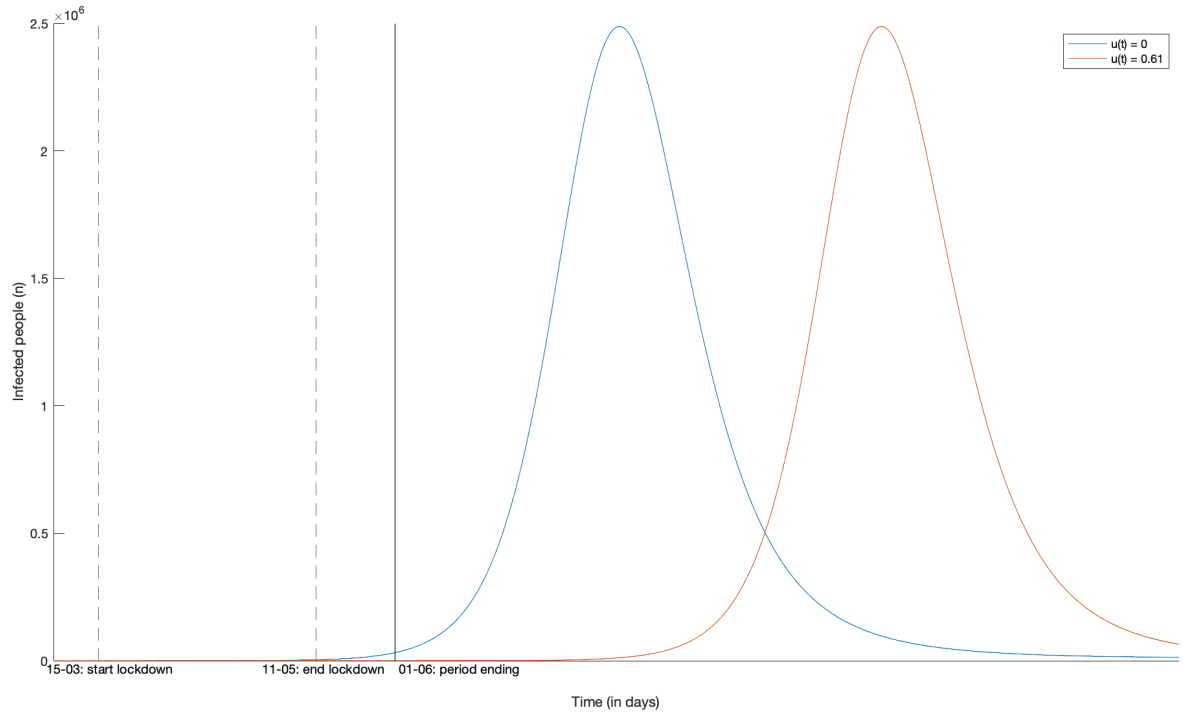

Figure 20: SEIRS-model over time with and without lockdown according to COVID-19 data.

To estimate the impact of the lockdown on the infection curve, the effectivity of the lockdown was determined in the same manner as done for the SIR-model (see Section 3.1.3). However, before determining this effectivity, it is crucial to also analyze whether the SEIRS-model with lockdown and parameter values as in Table 4 adequately describes the period prevalence of COVID-19 in the Netherlands (see Figure 1). This can be done by comparing the infection curves in the SEIRS-model with the period prevalence curve. We can also apply our priorknowledge on the theoretical SEIRS-model to say something about the current parameter fitting.

When comparing the infection curve with lockdown with the period prevalence curve, they are not similar since the maximum infection peak in the SEIRS-model is much higher, 2,487,700 in the SEIRS-model with lockdown versus 11,047 in the period prevalence curve (see Figure 1). However, it is because of the population size $N = 17,395,687$ that the infection peak can be equal to 2,487,700. This means that the population size chosen in the SEIRS-model is too great, which could indeed be the case, since COVID-19 hardly occurred in people up to and including 60 years of age and was mostly relevant for people older than 60 [20].

Additionally, the infection peak in the SEIRS-model is much later, 07-10-2020 in the SEIRS-model with lockdown versus 18-04-2020 in the period prevalence curve. As mentioned in Section 3.2.1, $E(1)$ is very small, and because of that, more time is needed for the infection curve to form the whole curve, which explains why both infection curves in Figure 20 are moved to the right and have their infection peak at a much later moment than there was in real life. Additionally, as mentioned in Section 3.2.1 as well, the lockdown was implemented at a time period in which the infection curve without lockdown did not even start increasing, which therefore also leads the infection curve to only move to the right instead of decreasing its maximum infection value.

Thus, even though the infection curve of the SEIRS-model does not necessarily look much like the period prevalence curve, the major differences were to be expected.

When considering the effectivity of the lockdown in the SEIRS-model, the effectivity of the lockdown is determined by comparing the maximum infection value without lockdown with the maximum infection value with lockdown. This is done for the period the infection curves with and without lockdown is fully visible instead for the period 01-03-2020 until 01-06-2020, since otherwise the maximum infection value for the infection curve with lockdown would be equal to 0, which does not accurately represent the true effect of the lockdown in the SEIRS-model. The effectivity of the lockdown is determined by the following quantity:

$$E_{SEIRS} := \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\%. \tag{8}$$

For the SEIRS-model with lockdown and the parameter values as in Table 4, we have $I_{max} = 2,487,700$ infections, and $I_{max+u(t)} = 2,487,600$ infections (see Figure 20). This means that the effectivity of the lockdown was equal to

$$E_{SEIRS} = \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\% = \left(1 - \frac{2,487,600}{2,487,700}\right) * 100\% = 0.004\%. \tag{9}$$

Thus, the effect of the lockdown for the SEIRS-model with parameter values as in Table 4 is negligible. However, the negligible effect of the lockdown on the maximum infection value is explicable: as mentioned in Section 3.2.1, the ratio between the initial value of infected people, $E(1)$, and the population size should be around $\frac{1}{10000}$ for the value of β and $u(t)$ to have an observable effect. However, the ratio for the SEIRS-model with lockdown and the parameters as in Table 4 is equal to $\frac{E(1)}{N} = \frac{12}{17395687} = 6.898 * 10^{-7}$, which is too small. Nevertheless, this makes sense, since the population size is in all likelihood too great. Also, the initial value of exposed people is equal to 12, but these were 12 infections reported in hospital that were exposed first [3]. It could very well be that the true initial value of exposed people was much higher due to underreporting. Also, Figure 21 shows a three-dimensional and contour plot of maximum infection values for all values of β and γ according to COVID-19 data (see Appendix T). From the figure and Section 3.2.2 we see that the contour plot looks like the contour plot without lockdown as in Figure 17, which means that the current values for β and therefore also γ have little or no impact on the infection values, which is indeed true (see Figure 20). Furthermore, from Section 3.2.1 we know that the values for μ, ω, σ, and α have considerable effect on the infection curve. Since changing μ and ω only cause the limit of the infection curve to go to a nonzero constant, it is not relevant to look at, since we only analyze the starting period of the infection curve and we do not consider the limit. Since Figure 20 shows that the infection curve with lockdown is moved too much to the right with respect to the period prevalence curve, and since we know the influence of α and σ from our simulations in Section 3.2.2, we now know that the value for α and σ were both too low. This makes sense, since we now know that people can also die from COVID-19, so $\alpha = 0$ was an incorrect assumption [3]. Also, we now know that the latency period instead of seven days could also have been shorter, namely five days [24]. In light of the above, the infection curve with lockdown in Figure 10 can be considered a representation of the impact of the ratio between the population size and initial infected people on the infection

curve, a lockdown that is implemented too early, and incorrect assumptions for values of α and γ, rather than the impact of implementing a lockdown on the infection curve.

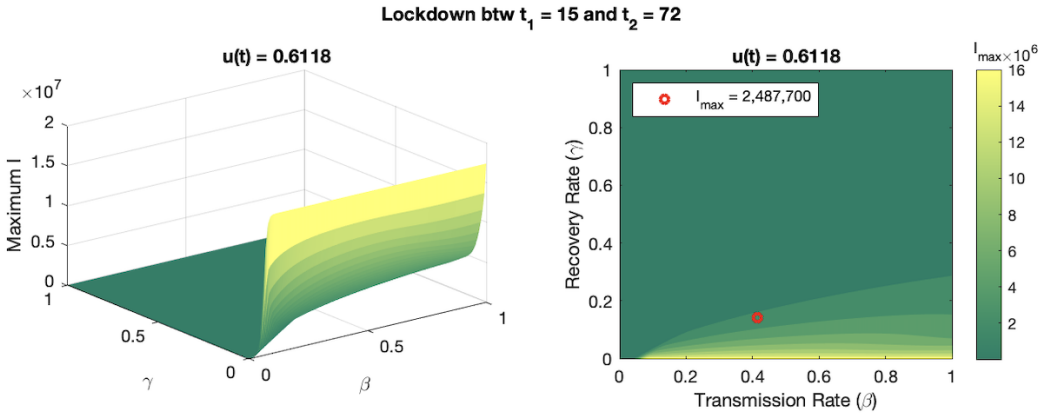

Figure 21: Three-dimensional (left) and contour (right) plots of maximum infection values for all values of β and γ according to COVID-19 data.

In consideration of the above, it is valuable to apply some more parameter fitting, to see whether we get a more adequate SEIRS-model and a more visible effect of the lockdown. For the infection curve in the SEIRS-model, the population size was adjusted to the population size of people aged 61 or older, which in March 2020 was equal to approximately 3.4 million people [16]. Also, the value for $E(1)$ was changed to $E(1) = 1200$ to obtain a proper ratio of $E(1)$ and N. Furthermore, the value of α was changed to $\alpha = \frac{1}{83*365}$ since people who died due to the consequences of COVID-19 were on average 83 years old [23]. Likewise, the value of σ was changed to $\sigma = \frac{1}{5}$, since literature has shown that the incubation time for COVID-19 could also be five days. Lastly, to see a better effect of the lockdown, instead of using the mean of $u(t)$, which was equal to $u(t) = 0.6118$, $u(t)$ is used as a time-dependent function in which the values for $u(t)$ change per day and are according to the mean lockdown curve as in Figure 9.

To see the impact of the change in parameter values seperately, the population size, the initial exposure value, and the values for α and β were first and at the same time changed, since visualisation of the impact of implementing a lockdown is dependent on a proper ratio between the population size, the initial exposure value, and the values for α and σ as mentioned in Section 3.2.2. These parameter values were all changed at the same time, since there otherwise would not be a proper ratio between the population size and the initial value of exposed people and the impact of a proper ratio and the visibility of the impact of the ratio between population size and initial value of exposed people is dependent on reasonable values for α and σ and vice versa. Figure 22 shows the SEIRS-model over time with and without lockdown with new parameter values for the population size and initial infection value. The figure shows that because of the decrease in population size the maximum infection values for both the infection curve with and without lockdown has decreased considerably. Also, both infection curves moved to the left because of the proper ratio between the new values of the population size and initial infection value. However, the impact of implementing a lockdown is still not very evident, since the

infection curve is still not enough during the period 01-03-2020 to 01-06-2020, which means that the lockdown period is too early and no effect of the lockdown on the infection curve is visible, as also explained in Section 3.2.1 (see Appendix M).

Figure 23 shows the SEIRS-model over time with and without lockdown with both new parameter values for population size, initial exposure value, other values for α and σ, as well as a time-dependent lockdown strength. Changing the lockdown strength to a time-dependent function upon changing the parameter values of the population size and initial infection value was done explicitly, since a decent visualisation of the impact of implementing a lockdown is dependent on a proper ratio between the population size and the initial exposure value, which would not have been the case using the population size and initial exposure value as in Table 4. However, the figure does not show an improvement in terms of maximum infection value. On the contrary, the maximum infection value has even increased compared to the infection curve without lockdown. The reason for this remains unclear, except that it must have something to do with the parameter fitting. On the other hand, the infection curve is moved more to the left, and therefore, the impact of β becomes more clear as explained in Section 3.2.2, as a result of which the maximum infection value could increase more easily and to a higher value. The influence of the lockdown is probably not seen because it was implemented too early to see the impact of the lockdown strength, despite its strength (see Appendix M).

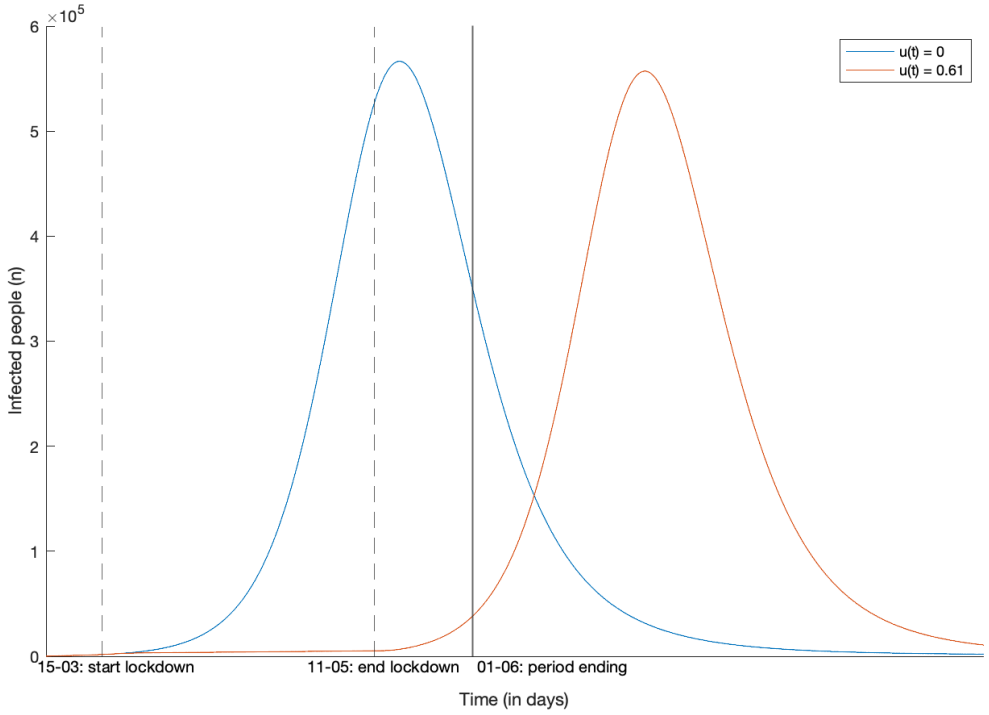

Figure 22: SEIRS-model over time with and without lockdown with new parameter values for N, $E(1)$, α, and σ

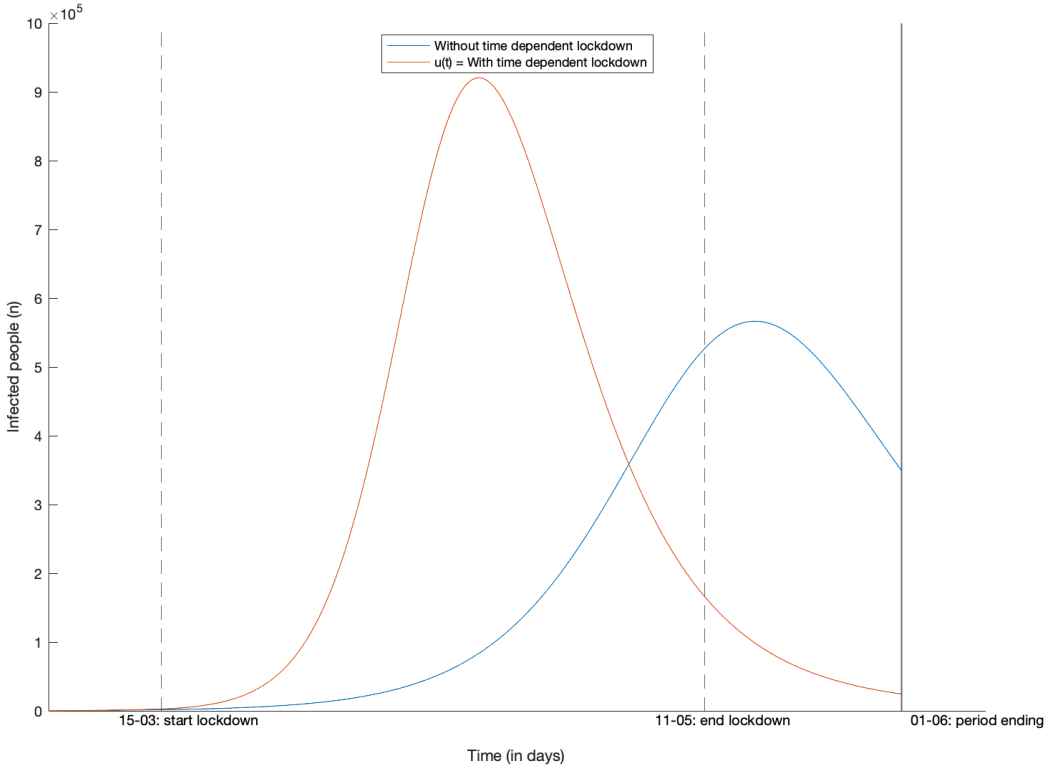

Figure 23: SEIRS-model over time with and without lockdown according to COVID-19 data

Applying the new parameter values into the SEIRS-model has not proven to provide a more adequate infection curve with and without lockdown, since it still did not show many resemblance with the period prevalence curve. On the contrary, it provided a worse description of reality. Therefore, the effectivity of the lockdown as calculated earlier can be considered the best effectivity for the SEIRS-model, which is equal to $E_{SEIRS} = 0.004\%$. Thus, the effect of the lockdown for the SEIRS-model with parameter values as in Table 4 is negligible, since a change of 0.004% in maximum infection values can be considered irrelevant. However, the theoretical infection curves as in Section 3.2.2 have shown that implementing a lockdown in a SEIRS-model certainly can be effective. However, this is dependent on correct parameter fitting, which can be concluded to not be possible for this SEIRS-model. However, it must be stated that the SEIRS-model has more parameter values which all can have an impact on each other, as a result of which small changes in parameter values can have a considerable impact on the infection curve, which makes interpretation of the influence of the parameter values more difficult.

4 Optimal starting day and duration

Until now, we tried to explore the effectivity of the lockdown implemented in the Netherlands from 15-03-2020 until 11-05-2020. To do this, we used the SIR- and SEIRS-models to simulate the period prevalence curve. When the infection curve with lockdown in the SIR- and SEIRS- model was considered adequate to describe the period prevalence curve, the effectivity was determined. It was concluded that implementing a lockdown indeed can be very effective to reduce the spread

of a virus. Now, given that implementing a lockdown is effective, it is interesting to analyze the influence of the starting day and duration of a lockdown, to see whether another starting day and/or duration could yield an even more effective lockdown. To see whether there potentially was a more optimal starting day or duration, first the SIR-model was used and then the SEIRS-model.

4.1 The SIR-model

A possibly better starting day and duration were determined with the SIR-model with the parameter values that yielded the infection curve with lockdown as in Figure 12 (i.e., $N = 3,400,000$; $I(1) = 1200$; and $u(t) = 0.6118$. See also Section 3.1.3). Because of a lack of sufficiently enough knowledge of MATLAB, $u(t)$ was used as a constant instead of the time-dependent function. However, the differences in maximum infection value between $u(t) = 0.6118$ and $u(t)$ was $609,020 - 582,480 = 26,540$, which can be considered small.

Figure 24 shows a three-dimensional and contour plot of the maximum infection values for all different starting days and durations, with a red circle representing the maximum infection value for the lockdown as it was in the Netherlands according to the SIR-model, with starting day 15-03-2020 ($t_1 = 15$) and a duration of 57 days, which was equal to $I_{max} = 609,020$ (see Appendix V). Figure 25 shows a plot of the maximum infection values for different starting days with constant duration of 57 days according to the COVID-19 data. According to Figure 24 and Figure 25, the maximum infection value with the current starting day and duration was already in the area where the maximum infection values are decreasing, which means that the current lockdown start and duration already was better than doing nothing (see Appendix W). Nonetheless, the maximum infection value could have been lower by *only* changing the starting day to $t_1 \approx 19$. Then, the maximum infection value would have been equal to $I_{max} = 204,360$. This makes sense, since the lockdown start would have been implemented more during the increase of the infection curve, and therefore, the lockdown has more impact on the maximum infection value, as also explained in Section 3.2.2. The effectivity of the lockdown in this case would have been equal to:

$$E_{SIR} = \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\% = \left(1 - \frac{204,360}{978,470}\right) * 100\% = 79.1\%. \tag{10}$$

Additionally, Figure 24 shows that changing *only* the duration of the lockdown to a shorter - and thus more acceptable - one, 49 days instead of 57, would also have yielded a lower maximum infection value, namely $I_{max} = 282,210$. The effectivity of the lockdown in this case would have been equal to:

$$E_{SIR} = \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\% = \left(1 - \frac{282,210}{978,470}\right) * 100\% = 71.2\%. \tag{11}$$

Thus, the above shows that despite the fact that the lockdown with starting day 15-03-2020 and duration of 57 days yielded a decrease in maximum infection values and yielded an effectivity

of 37.8%:

$$E_{SIR} = \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\% = \left(1 - \frac{609,020}{978,470}\right) * 100\% = 37.8\%, \tag{12}$$

when the lockdown would have been implemented a couple of days later or the duration would have been a week shorter, the maximum infection values would have been even less and the effectivity of the lockdown would have increased with 41.3% or 33.4% for changing the starting day and the duration respectively. It can be concluded that implementing a lockdown at the right moment for the right time can have an immense impact on the maximum infection values.

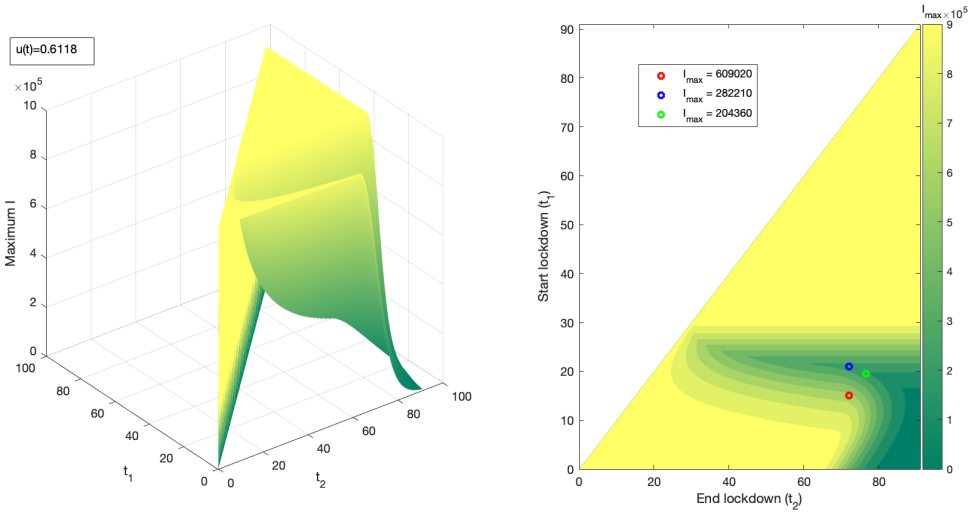

Figure 24: Three-dimensional and contour plot of maximum infection values for all possible time intervals according to COVID-19 data

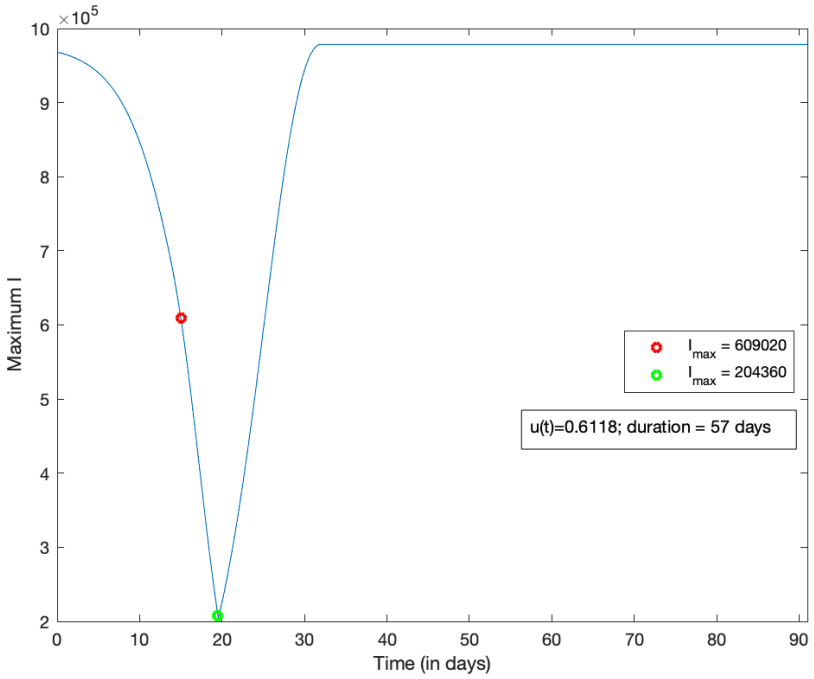

Figure 25: Plot of the maximum infection values for different starting days with constant duration of 57 days according to COVID-19 data

4.2 The SEIRS-model

A possibly better starting day and duration were determined with the SEIRS-model with the parameter values that yielded the infection curve with lockdown as in Figure 22 (i.e., $n = 3,400,000$; $E(1) = 1200$; $u(t) = 0.6118$; $\alpha = \frac{1}{83*365}$; and $\sigma = \frac{1}{5}$. See also Section 3.2.3). Again, because of a lack of sufficient knowledge on MATLAB $u(t)$ was used as a constant instead of the time-dependent function.

Figure 26 shows a three-dimensional and contour plot of the maximum infection values for all different starting days and durations, with a red circle representing the maximum infection value for the lockdown as it was in the Netherlands, with starting day 15-03-2020 ($t_1 = 15$) and a duration of 57 days, which was equal to $I_{max} = 552,630$ (see Appendix X). Figure 27 shows a plot of the maximum infection values for different starting days with constant duration of 57 days according to the COVID-19 data (see Appendix Y). According to Figure 26 and Figure 27, the maximum infection value with the current starting day and duration was already in the area where the maximum infection values are lowest, which means that the current lockdown start and duration were already optimal, despite the fact that the effectivity in this case is equal to:

$$E_{SEIRS} = \left(1 - \frac{I_{max+u(t)}}{I_{max}}\right) * 100\% = \left(1 - \frac{552,630}{566,460}\right) * 100\% = 2.4\%. \tag{13}$$

This means that the maximum infection values are less influenced by the lockdown start and/or duration.

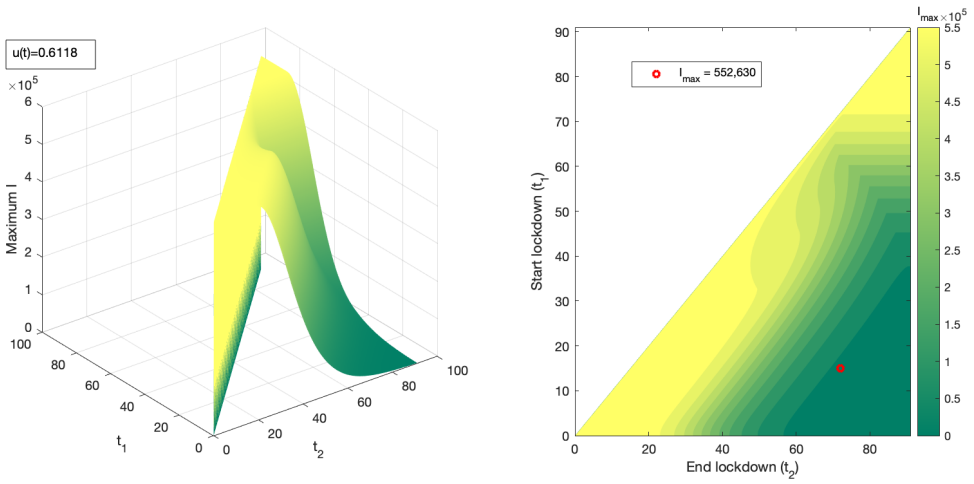

Figure 26: Three-dimensional and contour plot of maximum infection values for all possible time intervals according to COVID-19 data

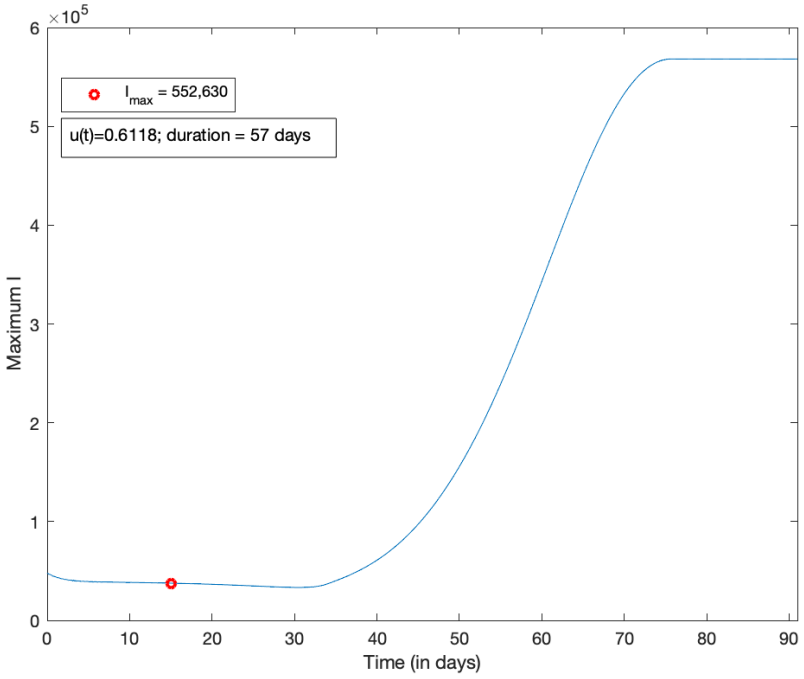

Figure 27: Plot of the maximum infection values for different starting days with constant duration of 57 days according to COVID-19 data

5 Pareto Efficiency

Now that we have also determined whether the lockdown in the Netherlands could have been more optimal in starting day or duration, the only thing left to determine is whether the lockdown at that time could be considered Pareto efficient.

5.1 Theoretical background

Pareto efficiency can be defined as a situation in which it is not possible to optimise one situation without it being at the expense of the other situation [6]. It is a concept from multi-objective optimisation theory, in which the aim is to optimise a problem with more than one objective [6]. The trade-off between the two objective functions is related to the level curves of the objective functions and its gradients:

Definition 1. *A solution is called Pareto efficient if and only if the gradient of one objective function is perpendicular to the level curve of the other objective function and both level curves have opposite directions.*

Definition 2. *The Pareto Front is the set of all Pareto efficient solutions.*

Here, it is desirable to minimize the maximum value of the infection curve while at the same time minimizing the duration and the strength of the lockdown. Therefore, the objective functions chosen were $I_{max} := \max I(t)$ and $u\,(t)_{tot} := \left(\int_{t_1}^{t_2} u(t)\,dt\right)$, which represent the maximum value of the infections and the surface of the duration and strength of the lockdown respectively. The Pareto Front could be simulated using the *gamultiobj* function in Matlab with $I\,(t)_{max}$ and $u\,(t)_{tot}$ as the objective functions (see Appendix 10 and 11).

5.2 Pareto efficiency for SIR- and SEIRS-model with arbitrary parameter values

Before applying actual data, it was valuable to analyze Pareto Fronts for different values of β and γ, since it can show us the impact of different values of the parameters. Figure 28 and Figure 29 show Pareto Fronts with I_{max} and u_{tot} as the objective functions for four different values of β (see Appendix Z and 7). The Pareto efficient solutions for all different values for β cause a vertical line for $I_{max} = 0$. This means that the Pareto efficient solutions are less dependent on u_{tot}, and therefore less dependent on the lockdown strength and duration, and more on the value for I_{max}. The four and three curves also show a more or less linear descending line, which means that as I_{max} increases in value, the values for u_{tot} decrease. This makes sense, since aiming for a low value of I_{max} is dependent on the choice of the lockdown strength and duration and the other way around.

Additionally, the Pareto Fronts show that when β increases in value, the number of Pareto efficient solutions for $I_{max} = 0$ decreases per value of β. However, the decrease is not linear, since the gap between $\beta = 0.3$ and $\beta = 0.5$ is much bigger than the gap between $\beta = 0.5$ and $\beta = 0.7$. Thus, for low values of β, aiming for a low I_{max} is longer independent of u_{tot} than for higher values of β. Lastly, as β increases in value, the starting value of descending increases, which makes sense since I_{max} is less dependent on the lockdown strength and duration for higher values of β.

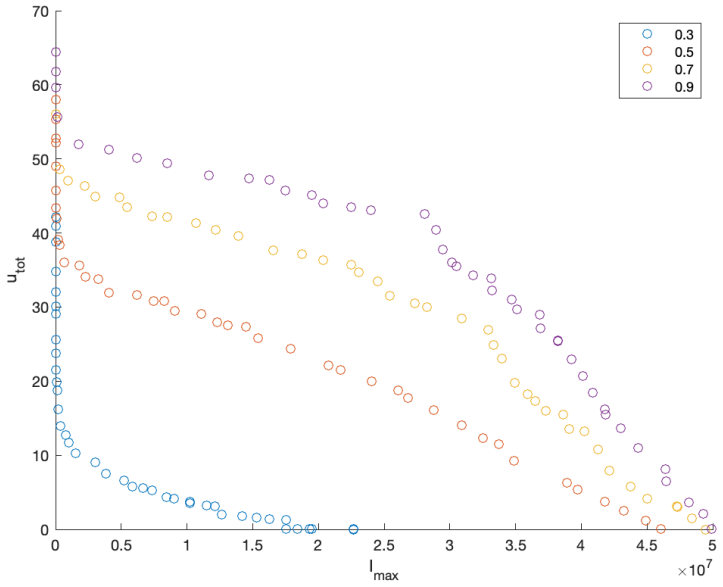

Figure 28: Pareto Front for SIR-model with $I\left(t\right)_{max}$ and $u\left(t\right)_{tot} = \int_{t_1}^{t_2} u(t)\, dt$ as the objective functions for different values of β

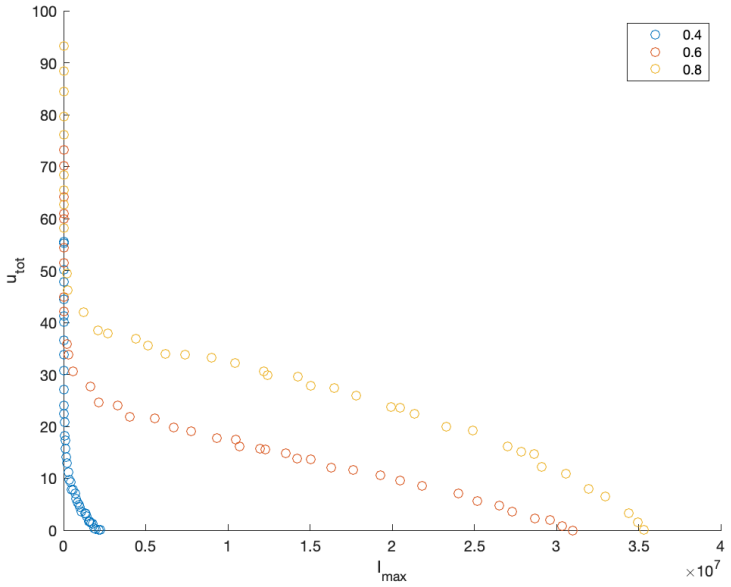

Figure 29: Pareto Front for SEIRS-model with $I\left(t\right)_{max}$ and $u\left(t\right)_{tot} = \int_{t_1}^{t_2} u(t)\, dt$ as the objective functions for different values of β

Figure 30 and Figure 31 show Pareto Fronts with I_{max} and u_{tot} as the objective functions for four different values of γ (see Appendix 8 and 9). The four and three curves also show a descending line, which means that as I_{max} increases in value, the values for u_{tot} decrease. This makes sense, since aiming for a low value of I_{max} is dependent on the choice of the lockdown strength and duration, and the other way around. Whereas the Pareto Fronts leave the y-axis earlier for bigger values for β, the Pareto Fronts leave the y-axis earlier for smaller values for γ. All curves show a vertical descent at $I_{max} = 0$ and later again an increase in descending pace, which

means that for $I_{max} = 0$ and at a certain value for I_{max} bigger changes in u_{tot} - and therefore in lockdown strength or duration - still imply Pareto efficient solutions for the same number of I_{max}.

The plots also show that for small changes in γ the Pareto Fronts differ greatly and for $\gamma \geq 0.5$ all Pareto efficient solutions are around (0,0) and could therefore not be plotted into the figure. The Pareto Fronts also show different curves, which means that the same change in size of γ does not imply that the change in Pareto Front is similar. This means that γ changes in impact per value of γ and impact increases when the value of γ decreases.

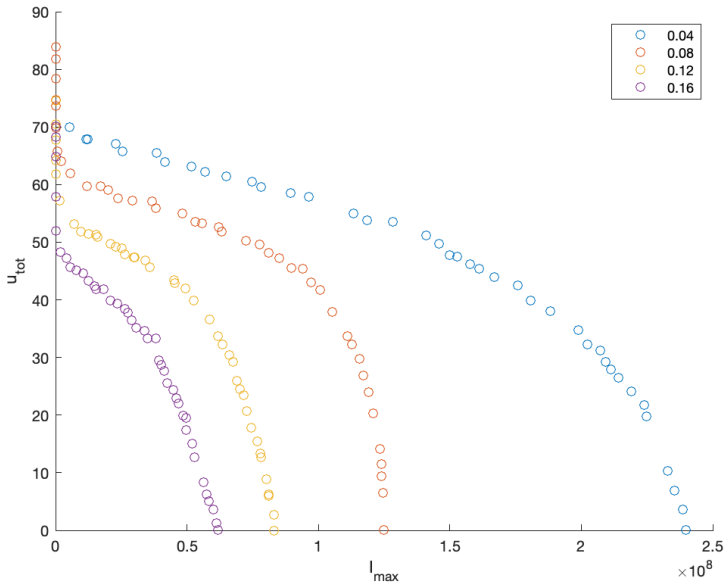

Figure 30: Pareto Front with I_{max} and $u\,(t)_{tot} = \int_{t_1}^{t_2} u(t)\,dt$ as the objective functions for different values of γ

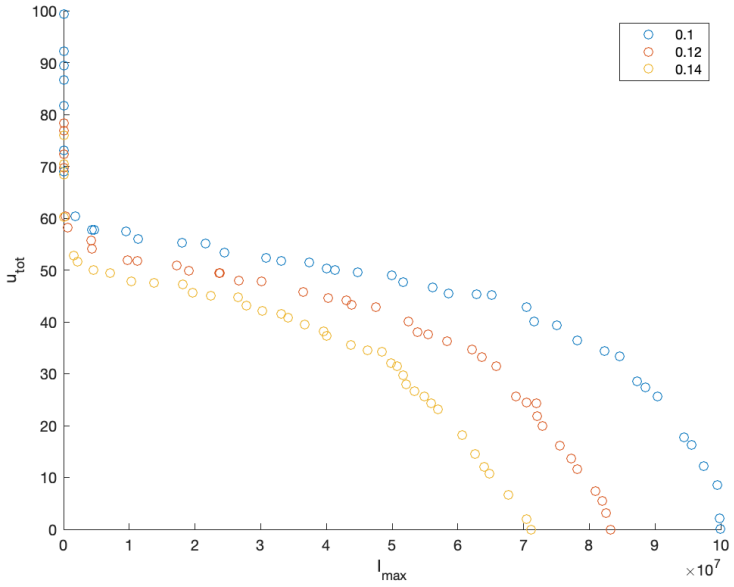

Figure 31: Pareto Front with I_{max} and $u\,(t)_{tot} = \int_{t_1}^{t_2} u(t)\,dt$ as the objective functions for different values of γ

5.3 Pareto efficiency for SIR- and SEIRS-model with parameter values from COVID-19 data

5.3.1 The SIR-model

For the SIR-model, Figure 32 shows the Pareto Front with I_{max} on the x-axis and u_{tot} on the y-axis. For the SIR-model, $u_{tot} = 57 * 0.6118 = 34.9$ and $I_{max} = 609,020$. This means that this combination of duration and lockdown strength was not optimal and could have been more optimal by reducing the lockdown strength or the duration (see Appendix 10).

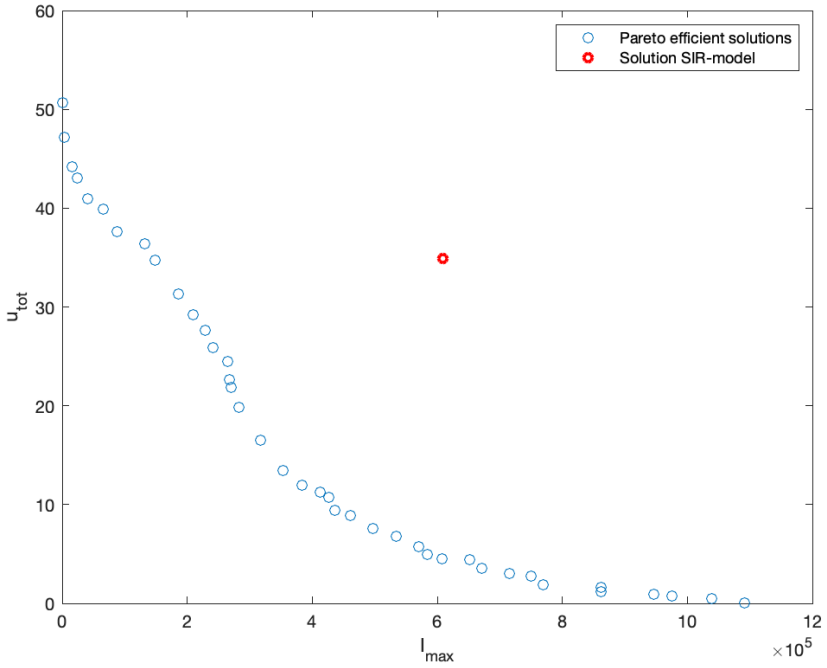

Figure 32: Pareto Front for SIR-model with $I\left(t\right)_{max}$ and $u\left(t\right)_{tot} = \int_{t_1}^{t_2} u(t)\, dt$ as the objective functions.

5.3.2 The SEIRS-model

For the SEIRS-model, Figure 33 shows the Pareto Front with I_{max} on the x-axis and u_{tot} on the y-axis. For the SEIRS-model, $u_{tot} = 57 * 0.6118 = 34.9$ and $I_{max} = 552630$. This means that this combination of duration and lockdown strength was not optimal and could have been improved by reducing the lockdown strength or the duration (see Appendix 11).

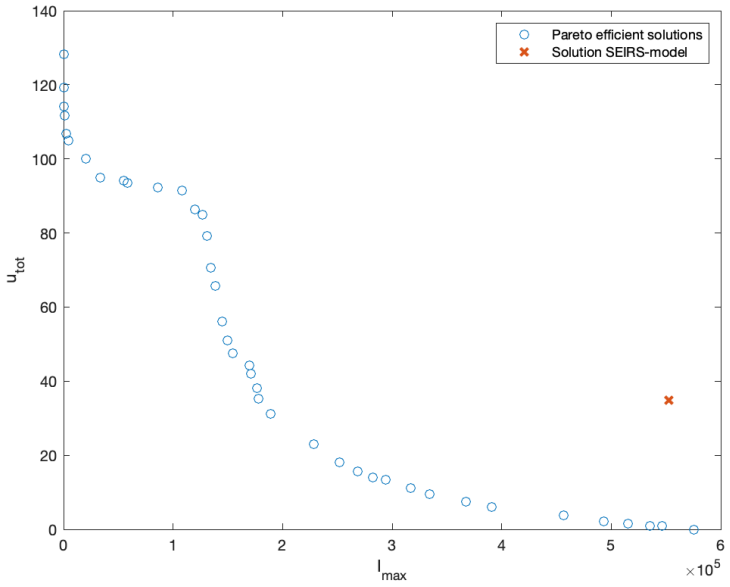

Figure 33: Pareto Front for SEIRS-model with I_{max} and $u_{tot} = \int_{t_1}^{t_2} u(t)\,dt$ as the objective functions

6 Conclusion

Many ways to reduce the spread of an infectious disease such as COVID-19 have been identified, both at the societal level as well as the individual level. One at the societal level is the introduction of lockdowns as a measure to reduce the spread of COVID-19. Lockdowns may be effective but have many drawbacks at the individual, societal and economic level. However, not much research has yet been done on the effectiveness of lockdowns as a measure to reduce the spread of infections. Therefore, this bachelor thesis focused on exploring the effectiveness of the first lockdown implemented in the Netherlands from 15-03-2020 until 11-05-2020. It was determined whether the lockdown at that time was Pareto efficient and whether there could have been a more optimal starting day and/or duration of the lockdown. A theoretical model describing reality was needed to determine the above. An important finding was that it proved quite difficult to find a theoretical model describing reality adequately. Additionally, the results implied that the lockdown in the Netherlands could have been imposed later and for a shorter duration. However, more research should be done on the influence of the parameter values on the theoretical models to form theoretical models that more realistically describe reality.

References

Alirol, E., Getaz, L., Stoll, B., Chappuis, F., & Loutan, L. (2011). Urbanisation and infectious diseases in a globalised world. *The Lancet Infectious Diseases*, *11*(2), 131–141. https://doi.org/10.1016/S1473-3099(10)70223-1

Ayouni, I., Maatoug, J., Dhouib, W., Zammit, N., Fredj, S. B., Ghammam, R., & Ghannem, H. (2021). Effective public health measures to mitigate the spread of COVID-19: A systematic review. *BMC Public Health*, *21*, 1015. https://doi.org/10.1186/s12889-021-11111-1

Balasko, Y. (1979). Budget-constrained Pareto-efficient allocations. *Journal of Economic Theory*, *21*(3), 359–379. https://doi.org/10.1016/0022-0531(79)90046-2

Bishop, T., Reinke, J., & Adams, T. (2011). Globalization: Trends and Perspectives. *Journal of International Business Research*.

Bjørnstad, O. N., Shea, K., Krzywinski, M., & Altman, N. (2020). The SEIRS model for infectious disease dynamics. *Nature Methods*, *17*(6), Article 6. https://doi.org/10.1038/s41592-020-0856-2

Bozkurt, A., Karakaya, K., Turk, M., Karakaya, Ö., & Castellanos-Reyes, D. (2022). The Impact of COVID-19 on Education: A Meta-Narrative Review. *Techtrends*, *66*(5), 883–896. https://doi.org/10.1007/s11528-022-00759-0

Britton, T., & Leskelä, L. (2023a). Optimal intervention strategies for minimizing total incidence during an epidemic. *SIAM Journal on Applied Mathematics*, *83*(2), 354–373. https://doi.org/10.1137/22M1504433

Britton, T., & Leskelä, L. (2023b). Optimal intervention strategies for minimizing total incidence during an epidemic. *SIAM Journal on Applied Mathematics*, *83*(2), 354–373. https://doi.org/10.1137/22M1504433

Burki, T. (2020). The indirect impact of COVID-19 on women. *The Lancet. Infectious Diseases*, *20*(8), 904–905. https://doi.org/10.1016/S1473-3099(20)30568-5

Cleaveland, S., Haydon, D. T., & Taylor, L. (2007). Overviews of Pathogen Emergence: Which Pathogens Emerge, When and Why? *Wildlife and Emerging Zoonotic Diseases: The Biology, Circumstances and Consequences of Cross-Species Transmission*, *315*, 85–111. https://doi.org/10.1007/978-3-540-70962-6_5

Cohen, J., & Rodgers, Y. van der M. (2020). Contributing factors to personal protective equipment shortages during the COVID-19 pandemic. *Preventive Medicine*, *141*, 106263. https://doi.org/10.1016/j.ypmed.2020.106263

COVID-19 dataset. (n.d.). Retrieved 8 May 2023, from https://data.rivm.nl/covid-19/

Cucinotta, D., & Vanelli, M. (2020). WHO Declares COVID-19 a Pandemic. *Acta Bio-Medica: Atenei Parmensis*, *91*(1), 157–160. https://doi.org/10.23750/abm.v91i1.9397

cycles, T. text provides general information S. assumes no liability for the information given being complete or correct D. to varying update, & Text, S. C. D. M. up-to-D. D. T. R. in the. (n.d.). *Topic: Coronavirus: impact on the global economy*. Statista. Retrieved 30 June 2023, from https://www.statista.com/topics/6139/covid-19-impact-on-the-global-economy/

Deaths from lower respiratory infections, by age. (n.d.). Retrieved 5 December 2022, from https://ourworldindata.org/grapher/deaths-from-lower-respiratory-infections-by-age

Di Domenico, L., Pullano, G., Sabbatini, C. E., Boëlle, P.-Y., & Colizza, V. (2020). Impact of lockdown on COVID-19 epidemic in Île-de-France and possible exit strategies. *BMC Medicine*, *18*, 240. https://doi.org/10.1186/s12916-020-01698-4

Forecasted global real GDP growth 2024. (n.d.-a). Statista. Retrieved 30 June 2023, from https://www.statista.com/statistics/1102889/covid-19-forecasted-global-real-gdp-growth/

Forecasted global real GDP growth 2024. (n.d.-b). Retrieved 20 June 2023, from https://www.statista.com/statistics/1102889/covid-19-forecasted-global-real-gdp-growth/

ILO: As job losses escalate, nearly half of global workforce at risk of losing livelihoods. (2020). http://www.ilo.org/global/about-the-ilo/newsroom/news/WCMS_743036/lang–en/index.htm

Jefferson, T., Del Mar, C. B., Dooley, L., Ferroni, E., Al-Ansary, L. A., Bawazeer, G. A., van Driel, M. L., Nair, N. S., Jones, M. A., Thorning, S., & Conly, J. M. (2011). Physical interventions to interrupt or reduce the spread of respiratory viruses. *The Cochrane Database of Systematic Reviews*, *2011*(7), CD006207. https://doi.org/10.1002/14651858.CD006207.pub4

Johnson, C. K., Hitchens, P. L., Pandit, P. S., Rushmore, J., Evans, T. S., Young, C. C. W., & Doyle, M. M. (2020). Global shifts in mammalian population trends reveal key predictors of virus spillover risk. *Proceedings of the Royal Society B: Biological Sciences*, *287*(1924), 20192736. https://doi.org/10.1098/rspb.2019.2736

Li, Y., & Nair, H. (2022). Trends in the global burden of lower respiratory infections: The knowns and the unknowns. *The Lancet Infectious Diseases*, *22*(11), 1523–1525. https://doi.org/10.1016/S1473-3099(22)00445-5

Lu, G., Zhang, Y., Zhang, H., Ai, J., He, L., Yuan, X., Bao, S., Chen, X., Wang, H., Cai, J., Wang, S., Zhang, W., & Xu, J. (n.d.). Geriatric risk and protective factors for serious COVID-19 outcomes among older adults in Shanghai Omicron wave. *Emerging Microbes & Infections*, *11*(1), 2045–2054. https://doi.org/10.1080/22221751.2022.2109517

Melikechi, O., Young, A. L., Tang, T., Bowman, T., Dunson, D., & Johndrow, J. (2022). Limits of epidemic prediction using SIR models. *Journal of Mathematical Biology*, *85*(4), 36. https://doi.org/10.1007/s00285-022-01804-5

Michaud, C. M. (2009). Global Burden of Infectious Diseases. *Encyclopedia of Microbiology*, 444–454. https://doi.org/10.1016/B978-012373944-5.00185-1

Ministerie van Volksgezondheid, W. en S. (n.d.-a). *Reproductiegetal | Coronadashboard | Rijksoverheid.nl*. Retrieved 8 May 2023, from https://coronadashboard.rijksoverheid.nl

Ministerie van Volksgezondheid, W. en S. (n.d.-b). *Sterfte | Coronadashboard | Rijksoverheid.nl*. Retrieved 30 June 2023, from https://coronadashboard.rijksoverheid.nl

Quesada, J. A., López-Pineda, A., Gil-Guillén, V. F., Arriero-Marín, J. M., Gutiérrez, F., & Carratala-Munuera, C. (2021). Incubation period of COVID-19: A systematic review and meta-analysis. *Revista Clinica Espanola*, *221*(2), 109–117. https://doi.org/10.1016/j.rceng.2020.08.002

Safiri, S., Mahmoodpoor, A., Kolahi, A.-A., Nejadghaderi, S. A., Sullman, M. J. M., Mansournia, M. A., Ansarin, K., Collins, G. S., Kaufman, J. S., & Abdollahi, M. (2023). Global burden of lower respiratory infections during the last three decades. *Frontiers in Public Health*, *10*, 1028525. https://doi.org/10.3389/fpubh.2022.1028525

Scholte, J. A. (2008). Defining Globalisation. *The World Economy*, *31*(11), 1471–1502. https://doi.org/10.1111/j.1467-9701.2007.01019.x

Skyrud, K., Telle, K., & Magnusson, K. (2021). Impacts of mild and severe COVID-19 on sick leave. *International Journal of Epidemiology*, *50*(5), 1745–1747. https://doi.org/10.1093/ije/dyab182

Statistiek, C. B. voor de. (n.d.-a). *Bevolkingspiramide* [Webpagina]. Centraal Bureau voor de Statistiek. Retrieved 30 June 2023, from https://www.cbs.nl/nl-nl/visualisaties/dashboard-bevolking/bevolkingspiramide

Statistiek, C. B. voor de. (n.d.-b). *Bevolkingsteller* [Webpagina]. Centraal Bureau voor de Statistiek. Retrieved 30 June 2023, from https://www.cbs.nl/nl-nl/visualisaties/dashboard-bevolking/bevolkingsteller

Stok, F. M., Bal, M., Yerkes, M. A., & de Wit, J. B. F. (2021). Social Inequality and Solidarity in Times of COVID-19. *International Journal of Environmental Research and Public Health*, *18*(12), 6339. https://doi.org/10.3390/ijerph18126339

The impact of COVID-19 on mental, neurological and substance use services. (n.d.). Retrieved 1 May 2023, from https://www.who.int/publications-detail-redirect/978924012455

WHO Coronavirus (COVID-19) Dashboard. (n.d.). Retrieved 1 May 2023, from https://covid19.who.int

Woc-Colburn, L., & Godinez, D. (2022). Lockdown as a public health measure. *COVID-19 Pandemic*, 133–136. https://doi.org/10.1016/B978-0-323-82860-4.00013-6

World Economic Outlook, October 2020: A Long and Difficult Ascent. (n.d.). Retrieved 1 May 2023, from https://www.imf.org/en/Publications/WEO/Issues/2020/09/30/world-economic-outlook-october-2020

Wu, Y.-C., Chen, C.-S., & Chan, Y.-J. (2020). The outbreak of COVID-19: An overview. *Journal of the Chinese Medical Association*, *83*(3), 217–220. https://doi.org/10.1097/JCMA.0000000000000270

Zaken, M. van A. (2022, June 17). *Coronavirus tijdlijn—Rijksoverheid.nl* [Onderwerp]. Ministerie van Algemene Zaken. https://www.rijksoverheid.nl/onderwerpen/coronavirus-tijdlijn

(N.d.). Retrieved 30 June 2023, from https://covid19.who.int/data

Appendices

A Appendix A

```
clear;
close all

% Read reproductionnumbers data
data = readtable("reproductiegetallen.csv");

% Select columns to use
land = data.location;
datum = data.date;
reproductiegetal = data.reproduction_rate;
infectiesnieuw = data.new_cases;
cumulatief = data.total_cases;

% Make table
tabel = table(land, datum, reproductiegetal, infectiesnieuw,
   cumulatief);
tabel.land = cellstr(tabel.land);
land = "Netherlands";
index = tabel.land == land;

% Create column with specific country only
nederland = tabel.land(index);
datums = tabel.datum(index);
reproductiegetal_nl = tabel.reproductiegetal(index);
infectiesnieuw_nl = tabel.infectiesnieuw(index);
cumulatief_nl = tabel.cumulatief(index);
nederlandtabel = table(nederland, datums, reproductiegetal_nl,
   infectiesnieuw_nl, cumulatief_nl);

% Select data from 01-03-2020 until 01-06-2020
startdatum = datetime('2020-03-06');
einddatum4 = datetime('2020-06-01');
beginlockdown = datetime('2020-03-15');
eindlockdown = datetime('2020-05-11');
maartjuni_nederland = nederlandtabel(nederlandtabel.datums >=
   startdatum & nederlandtabel.datums <= einddatum4, :);

plot(maartjuni_nederland.datums, maartjuni_nederland.
```

```
   reproductiegetal_nl);
plot(maartjuni_nederland.datums, maartjuni_nederland.
   cumulatief_nl);

% NEW CASES
% Calculate sum of current values and last 10 values
maartjuni_nederland.Goede = zeros(size(maartjuni_nederland.
   infectiesnieuw_nl));
maartjuni_nederland.Goede(1:9) = maartjuni_nederland.
   infectiesnieuw_nl(1:9);
for i = 10:numel(maartjuni_nederland.infectiesnieuw_nl)
    maartjuni_nederland.Goede(i) = sum(maartjuni_nederland.
       infectiesnieuw_nl(i-9:i));
end

% Plot of period prevalence curve
plot(maartjuni_nederland.datums, maartjuni_nederland.Goede)
xlim([datetime('01-03-2020', 'InputFormat', 'dd-MM-yyyy')
   datetime('01-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel('Dates')
ylabel ('Infected people (n)')
hold on;
text(beginlockdown, 0, '15-03', 'VerticalAlignment', 'top', '
   HorizontalAlignment', 'center');
text(eindlockdown, 0, '11-05', 'VerticalAlignment', 'top', '
   HorizontalAlignment', 'center');
hold off;
```

B Appendix B

```
clear;
close all;

% Length of simulation (in days)
tmax = 100;
% Time step (in days)
dt = 0.01;
% Time values (in days)
t=0:dt:tmax;

% Create stepfunction
u = zeros(size(t));
u(t>=0 & t<15) = 0;
```

```
u(t>=15 & t<72) = 0.75;
u(t>=72 & t<=92) = 0;

% Plot of exponential lockdown
plot(t,u)
xlabel('Time (in days)')
ylabel('Lockdown strength')

begindag = 0;
einddag = 92;
beginlockdown = 15;
eindlockdown = 72;
simulatieeinde = 92;

xticks([]);
hold on;
text(begindag, 0, '01-03', 'VerticalAlignment', 'bottom', '
   HorizontalAlignment', 'center');
text(einddag, 0, '01-06', 'VerticalAlignment', 'bottom', '
   HorizontalAlignment', 'center');
text(beginlockdown, 0, '15-03', 'VerticalAlignment', 'bottom', '
   HorizontalAlignment', 'center');
text(eindlockdown, 0, '11-05', 'VerticalAlignment', 'bottom', '
   HorizontalAlignment', 'center');
hold off;
```

C Appendix C

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 100;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
I = zeros(1, nt); % Infected values
```

```
R = zeros(1, nt); % Recovered values
S(1) = P - I0;    % Initial value of S
I(1) = I0;        % Initial value of I
R(1) = 0;         % Initial value of R

% Calculations
beta = 0.6;                  % Transmission rate (per day)
gamma = 0.2;                 % Recovery rate (per day)
lockdown_values = [0, 0.75]; % Lockdown strength values

figure; % Create a new figure for the plot

hold on; % Enable holding the plot to add multiple lines

% Create SIR model for each lockdown value
for i = 1:length(lockdown_values)
    lockdown = lockdown_values(i);

    % Create step function for u
    u = zeros(1, nt);
    u(t >= 0 & t < 20) = 0;
    u(t >= 20 & t < 40) = lockdown;
    u(t >= 40) = 0;

    % Reset initial values for each lockdown value
    S(1) = P - I0;
    I(1) = I0;
    R(1) = 0;

    % Create SIR model
    for j = 2:nt
        dS = (-beta*(1-u(j))*I(j-1)*S(j-1)/P)*dt;              %
           Change in S
        S(j) = S(j-1) + dS;                                    %
           Current S value
        if S(j) < 0; S(j) = 0; end

        dI = (beta*(1-u(j))*I(j-1)*S(j-1)/P-gamma*I(j-1))*dt; %
           Change in I
        I(j) = I(j-1)+dI;                                      %
           Current S value
        if I(j) < 0; I(j) = 0; end
```

```
        dR = (gamma*I(j-1))*dt;                                    %
            Change in R
        R(j) = R(j-1) + dR;                                        %
            Current R value
    end
    grootste = max(I(:));

    % Plot of SIR model
    plot(t, I, 'DisplayName', sprintf('Lockdown: %.2f', lockdown)
        );
end

hold off; % Release the hold on the plot

% Add titles, xlabel, ylabel, legends
xlabel('Time (in days)')
ylabel ('Infected people (n)')
legenda1 = legend('u(t) = 0', 'u(t) = 0.75');
set(legenda1, 'Position', [0.75, 0.78, 0.15, 0.05])
legendatekst1 = 'Lockdown start: 20';
legendatekst2 = 'Lockdown end: 40';
legendapositie = [0.72, 0.85, 0.18, 0.06]; %x,y,breedte,hoogte
legenda2 = annotation('textbox', legendapositie, 'String', {
    legendatekst1, legendatekst2}, 'FontSize', 9);
```

D Appendix D

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 100;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;         % Time values (in days)
nt = length(t);      % Number of timesteps
S = zeros(1, nt);    % Susceptible values
I_b = zeros(1, nt); % Infected values
```

```
R = zeros(1, nt);    % Recovered values
S(1) = P - I0;       % Initial value of S
I_b(1) = I0;         % Initial value of I
R(1) = 0;            % Initial value of R

% Calculations
beta = 0.6;       % Transmission rate (per day)
gamma = 0.2;      % Recovery rate (per day)
lockdown = 0.75;

% Define lockdown start and end values
startdays = [14, 18, 20, 22];
enddays = [34, 38, 40, 42];
aantal_plots = length(startdays);

figure; % Create a new figure for the plot

hold on; % Enable holding the plot to add multiple lines

% Create SIR model for each starting day and ending day
for i = 1:aantal_plots
    startday = startdays(i);
    endday = enddays(i);

    % Create step function for u
    u = zeros(1, nt);
    u(t >= 0 & t < startday) = 0;
    u(t >= startday & t < endday) = lockdown;
    u(t >= endday) = 0;

    % Reset initial values for each plot
    S(1) = P - I0;
    I_b(1) = I0;
    R(1) = 0;

    % Create SIR model
    for j = 2:nt
        dS = (-beta*(1-u(j))*I_b(j-1)*S(j-1)/P)*dt;              %
            Change in S
        S(j) = S(j-1) + dS;                                      %
            Current S value
        if S(j) < 0; S(j) = 0; end
```

```
        dI = (beta*(1-u(j))*I_b(j-1)*S(j-1)/P-gamma*I_b(j-1))*dt;
            % Change in I
        I_b(j) = I_b(j-1)+dI;
                                                % Current S value
        if I_b(j) < 0; I_b(j) = 0; end

        dR = (gamma*I_b(j-1))*dt;                                    %
            Change in R
        R(j) = R(j-1) + dR;                                          %
            Current R value
    end

    % Plot of SIR model
    plot(t, I_b, 'DisplayName', sprintf('Starting day: %d; Ending
        day: %d', startday, endday));
end

hold off; % Release the hold on the plot

% Add xlabel, ylabel, and legends to plot
xlabel('Time (in days)')
ylabel ('Infected people (n)')
legend('Lockdown start: 14; Lockdown end: 34', 'Lockdown start:
   18; Lockdown end: 38', 'Lockdown start: 20; Lockdown end: 40',
   'Lockdown start: 22; Lockdown end: 44')
legendatekst1 = 'u(t) = 0.75';
legendapositie = [0.79, 0.72, 0.1, 0.04]; %x,y,breedte,hoogte
legenda2 = annotation('textbox', legendapositie, 'String', {
   legendatekst1}, 'FontSize', 9);
```

E Appendix E

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 100;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)
```

```
% Define beta en gamma values
beta_values = 0:dt:1;
gamma_values = 0:dt:1;

% Initialize arrays for maximum I values
max_I_values = zeros(length(gamma_values), length(beta_values));

% Loop over beta and gamma values
for i = 1:length(beta_values)
    for j = 1:length(gamma_values)

        % Initialize values
        t=0:dt:tmax;       % Time values (in days)
        nt = length(t);    % Number of timesteps
        S = zeros(1, nt); % Susceptible values
        I = zeros(1, nt); % Infected values
        R = zeros(1, nt); % Recovered values
        S(1) = P - I0;     % Initial value of S
        I(1) = I0;         % Initial value of I
        R(1) = 0;          % Initial value of R

        % Calculations
        beta = beta_values(i);       % Transmission rate (per day)
        gamma = gamma_values(j);     % Recovery rate (per day)
        lockdown = 0;                % Lockdown strength

        % Create step function for u
        u = zeros(1, nt);
        u(t >= 0 & t < 20) = 0;
        u(t >= 20 & t < 40) = lockdown;
        u(t >= 40) = 0;

        % Create SIR model
        for k = 2:nt
            dS = (-beta*(1-u(k))*I(k-1)*S(k-1)/P)*dt;
                              % Change in S
            S(k) = S(k-1) + dS;
                                                  % Current S value
            if S(k) < 0; S(k) = 0; end

            dI = (beta*(1-u(k))*I(k-1)*S(k-1)/P-gamma*I(k-1))*dt;
                % Change in I
```

```
            I(k) = I(k-1)+dI;
                                                    % Current S
                value
            if I(k) < 0; I(k) = 0; end

            dR = (gamma*I(k-1))*dt;
                                                % Change in R
            R(k) = R(k-1) + dR;
                                                    % Current R value
        end

        % Find maximum I value and save it
        max_I = max(I);
        max_I_values(j,i) = max_I;
    end
end

% Create meshgrid for beta and gamma values
[BETA, GAMMA] = meshgrid(beta_values, gamma_values);

% Create 8 figures: 4 contour, 4 surface
figure;

% Plot 3D surface 1
subplot(4,2,1);
surf(BETA, GAMMA, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
title('u(t) = 0');

% Create contour plot 1
subplot(4,2,2);
contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
colormap summer;
colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
cb= colorbar;
title(cb, 'I_{max}');
```

```
title('u(t) = 0');

% Plot 3D surface 2
subplot(4,2,3);
surf(BETA, GAMMA, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
zlim([0,5e6]);
xlim([0, 1])
ylim([0, 1]);
clim([0, 5e6]);
title('u(t) = 0');

% Create contour plot 2
subplot(4,2,4);
contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
colormap summer;
colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
zlim([0,5e6]);
xlim([0,1])
ylim([0, 1]);
clim([0, 5e6]);
colorbar('Ticks', 0:1e6:5e6);
cb= colorbar;
title(cb, 'I_{max}');
title('u(t) = 0');

% Plot 3D surface 3
subplot(4,2,5);
surf(BETA, GAMMA, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
zlim([0,10e5]);
xlim([0, 1])
```

```
ylim([0, 1]);
clim([0, 10e5]);
title('u(t) = 0');


% Create contour plot 3
subplot(4,2,6);
contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
colormap summer;
colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
zlim([0,10e5]);
xlim([0,1])
ylim([0, 1]);
clim([0, 10e5]);
colorbar('Ticks', 0:2e5:10e5);
cb= colorbar;
title(cb, 'I_{max}');
title('u(t) = 0');

% Plot 3D surface 4
subplot(4,2,7);
surf(BETA, GAMMA, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
zlim([0,20000]);
xlim([0, 1])
ylim([0, 1]);
clim([0, 20000]);
title('u(t) = 0');

% Create contour plot 4
subplot(4,2,8);
contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
colormap summer;
colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
```

```
zlim([0,20000]);
xlim([0,1])
ylim([0, 1]);
clim([0, 20000]);
colorbar('Ticks', 0:5000:20000);
cb= colorbar;
title(cb, 'I_{max}');
title('u(t) = 0');
sgtitle('Lockdown btw t_1 = 20 and t_2 = 40', 'FontSize', 11, '
   FontWeight', 'bold')
```

F Appendix F

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 100;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Define beta and gamma values
beta_values = 0:dt:1;
gamma_values = 0:dt:1;

% Define lockdown values
lockdown_values = [0.35, 0.55, 0.75];

% Create figure
figure;

for l = 1:length(lockdown_values)
    % Initialize arrays for maximum I values
    max_I_values = zeros(length(gamma_values), length(beta_values
       ));

    % Loop over beta en gamma waarden
    for i = 1:length(beta_values)
        for j = 1:length(gamma_values)

            % Initialize values
```

```
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
I(1) = I0;         % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = beta_values(i);           % Transmission rate (
   per day)
gamma = gamma_values(j);         % Recovery rate (per
   day)
lockdown = lockdown_values(l);  % Lockdown strength

% Create step function for u
u = zeros(1, nt);
u(t >= 0 & t < 20) = 0;
u(t >= 20 & t < 40) = lockdown;
u(t >= 40) = 0;

% Create SIR model
for k = 2:nt
    dS = (-beta*(1-u(k))*I(k-1)*S(k-1)/P)*dt;
                    % Change in S
    S(k) = S(k-1) + dS;
                                          % Current S
       value
    if S(k) < 0; S(k) = 0; end

    dI = (beta*(1-u(k))*I(k-1)*S(k-1)/P-gamma*I(k-1))
       *dt; % Change in I
    I(k) = I(k-1)+dI;
                                            % Current S
        value
    if I(k) < 0; I(k) = 0; end

    dR = (gamma*I(k-1))*dt;
                                        % Change in R
    R(k) = R(k-1) + dR;
                                          % Current R
```

```
                value
            end

            % Find maximum I value and save it
            max_I = max(I);
            max_I_values(j,i) = max_I;
        end
    end

    % Create meshgrid for beta en gamma waarden
    [BETA, GAMMA] = meshgrid(beta_values, gamma_values);

    % Plot 3D surface
    subplot(3,2,l*2-1);
    surf(BETA, GAMMA, max_I_values);
    shading flat;
    colormap summer;
    xlabel('\beta');
    ylabel('\gamma');
    zlabel('Maximum I');
    title(['u(t) = ', num2str(lockdown_values(l))]);

    % Create contour plot
    subplot(3,2,l*2);
    contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
    colormap summer;
    cb= colorbar;
    title(cb, 'I_{max}');
    xlabel('Transmission Rate (\beta)');
    ylabel('Recovery Rate (\gamma)');
    title(['u(t) = ', num2str(lockdown_values(l))]);
    sgtitle('Lockdown btw t_1 = 20 and t_2 = 40', 'FontSize', 11,
        'FontWeight', 'bold')
end
```

G Appendix G

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
```

```
I0 = 1200;       % Initial infected population
tmax = 91;       % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;             % Time values (in days)
nt = length(t);          % Number of timesteps
S = zeros(1, nt);        % Susceptible values
I_nieuw = zeros(1, nt); % Infected values
R = zeros(1, nt);        % Recovered values
S(1) = P - I0;           % Initial value of S
I_nieuw(1) = I0;         % Initial value of I
R(1) = 0;                % Initial value of R

% Calculations
beta = 1;                % Transmission rate (per day)
gamma = 1/7;             % Recovery rate (per day)

% REPRODUCTIONNUMBER SIR-model
% load tables from other file
load('tabelIT.mat', 'maartjuni_italie');
load('tabelFR.mat', 'maartjuni_frankrijk');
load('tabelSP.mat', 'maartjuni_spanje');

reproductiegetallen = table(maartjuni_frankrijk.datums(3:93),
   maartjuni_italie.reproductiegetal_it(9:99), maartjuni_frankrijk
   .reproductiegetal_fr(3:93), maartjuni_spanje.reproductiegetal_s
   (1:91));
reproductiegetallen.Properties.VariableNames{'Var1'} = 'Datums';
reproductiegetallen.Properties.VariableNames{'Var2'} = 'Italie';
reproductiegetallen.Properties.VariableNames{'Var3'} = 'Frankrijk
   ';
reproductiegetallen.Properties.VariableNames{'Var4'} = 'Spanje';
gemiddelde = mean(reproductiegetallen{:, {'Italie', 'Frankrijk',
   'Spanje'}}, 2);
reproductiegetallen.Gemiddelde = gemiddelde;
reproductiegetallen.Betagetal = reproductiegetallen.Gemiddelde*
   gamma;

R0_IT = mean(maartjuni_italie.reproductiegetal_it(1:15));
R0_FR = mean(maartjuni_frankrijk.reproductiegetal_fr(1:16));
R0_SP = mean(maartjuni_spanje.reproductiegetal_s(1:11));
```

```
beta0_IT = R0_IT*gamma;
beta0_FR = R0_FR*gamma;
beta0_SP = R0_SP*gamma;

beta0 = (beta0_IT + beta0_FR + beta0_SP)/3;

% PLOT OF REPRODUCTIONNUMBERS OF EACH COUNTRY AND MEAN
   REPRODUCTION LINE
figure;
plot(reproductiegetallen.Datums, reproductiegetallen.Italie);
hold on;
plot(reproductiegetallen.Datums, reproductiegetallen.Frankrijk);
plot(reproductiegetallen.Datums, reproductiegetallen.Spanje);
plot(reproductiegetallen.Datums, reproductiegetallen.Gemiddelde,
   'LineWidth', 2);
hold off;

xlim([datetime('01-03-2020', 'InputFormat', 'dd-MM-yyyy')
   datetime('01-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel("Dates");
ylabel("Reproduction numbers");
legend('Italy', 'France', 'Spain', 'Mean', 'location', 'northeast
   ');

saveas(gcf, 'reproductiegetallenITFRSP.png');

% LOCKDOWNS

lockdown_IT = 1-((reproductiegetallen.Italie(7:63)*0.1429)/beta0)
   ;
lockdown_FR = 1-((reproductiegetallen.Frankrijk(15:70)*0.1429)/
   beta0);
lockdown_SP = 1-((reproductiegetallen.Spanje(12:91)*0.1429)/beta0
   );

u_IT = mean(lockdown_IT);
u_FR = mean(lockdown_FR);
u_SP = mean(lockdown_SP);
u0 = (u_IT + u_FR + u_SP)/3;

lockdown_IT2 = 1-((reproductiegetallen.Italie(13:70)*0.1429)/
```

```
   beta0);
lockdown_FR2 = 1-((reproductiegetallen.Frankrijk(13:70)*0.1429)/
   beta0);
lockdown_SP2 = 1-((reproductiegetallen.Spanje(13:70)*0.1429)/
   beta0);
gemiddeldelockdown = 1-((reproductiegetallen.Gemiddelde(13:70)
   *0.1429)/beta0);

% PLOT OF LOCKDOWNS
figure;
plot(reproductiegetallen.Datums(13:70), lockdown_IT2);
hold on;
plot(reproductiegetallen.Datums(13:70), lockdown_FR2);
plot(reproductiegetallen.Datums(13:70), lockdown_SP2);
plot(reproductiegetallen.Datums(13:70), gemiddeldelockdown, '
   LineWidth', 2);
hold off;

xlim([datetime('01-03-2020', 'InputFormat', 'dd-MM-yyyy')
   datetime('01-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel("Dates");
ylabel("Lockdownstrength");
legend('Italy', 'France', 'Spain', 'Mean', 'location', 'east');

saveas(gcf, 'lockdowngetallenITFRSP.png');

% Make column to use in SIR-model for beta and u(t)
goede = zeros(91,1);
goede(1:12) = beta0;
goede(13:70) = beta0*gemiddeldelockdown;
goede(71:91) = beta0;

reproductiegetallen.betametlockdown = goede;

% LOCKDOWNSTRENGTH SIR-model
load('lockdowntabelIT.mat', 'lockdowntabel_it');
load('lockdowntabelFR.mat', 'lockdowntabel_fr');
load('lockdowntabelSP.mat', 'lockdowntabel_s');

herhaling = reshape(repmat(reproductiegetallen.betametlockdown,
   1,100)', [], 1);
```

```
beginlockdown = 15;  % Start lockdown
eindlockdown = 72;   % End lockdown
beginsimulatie = 1;  % Beginday
eindsimulatie = 92;  % Endday

u(1) = herhaling(1);

% Create SIR model
for i = 2:nt
    u(i) = herhaling(i-1);

    dS = (-beta*(u(i))*I_nieuw(i-1)*S(i-1)/P)*dt;
                          % Change in S
    S(i) = S(i-1) + dS;
                                                  % Current S
       value
    if S(i) < 0; S(i) = 0; end

    dI = (beta*(u(i))*I_nieuw(i-1)*S(i-1)/P-gamma*I_nieuw(i-1))*
       dt; % Change in I
    I_nieuw(i) = I_nieuw(i-1)+dI;
                                          % Current I value
    if I_nieuw(i) < 0; I_nieuw(i) = 0; end

    dR = (gamma*I_nieuw(i-1))*dt;
                                          % Change in R
    R(i) = R(i-1) + dR;
                                                  % Current R
       value
end

grootste = max(I_nieuw(:));              % Maximum value of I
disp(grootste)
oppervlakte = trapz(t, I_nieuw(:));      % Surface under I graph
disp(oppervlakte)

% Plot of I graph from other file
plot(t,I_nieuw)
save('coronanederland_verbeterde_parameters.mat', 't', 'I_nieuw')
   ;
xlabel('Time (in days)')
ylabel ('Infected people (n)')
```

H Appendix H

```
clear;
close all

% Read reproductionnumbers data
data = readtable("reproductiegetallen.csv");

% Select columns to use
land = data.location;
datum = data.date;
reproductiegetal = data.reproduction_rate;

% Make table
tabel = table(land, datum, reproductiegetal);
tabel.land = cellstr(tabel.land);
land = "France";
index = tabel.land == land;

% Create new column with specific country only
frankrijk = tabel.land(index);
datums = tabel.datum(index);
reproductiegetal_fr = tabel.reproductiegetal(index);
frankrijktabel = table(frankrijk, datums, reproductiegetal_fr);

% Select data from 01-03-2020 until 01-06-2020
startdatum = datetime('2020-03-01');
einddatum4 = datetime('2020-06-01');
beginlockdown = datetime('2020-03-17');
maartjuni_frankrijk = frankrijktabel(frankrijktabel.datums >=
   startdatum & frankrijktabel.datums <= einddatum4, :);

plot(maartjuni_frankrijk.datums, maartjuni_frankrijk.
   reproductiegetal_fr);

% Calculate beta0 and u(t)
gamma = 1/7;
sigma = 1/7;
mu = 1/(76*365);
reproductie0 = mean(maartjuni_frankrijk.reproductiegetal_fr(1:18)
   );
beta0 = reproductie0*gamma;
```

```
maartjuni_frankrijk.lockdownbetagetal = maartjuni_frankrijk.
    reproductiegetal_fr*gamma;

reproductieseirs0 = mean(maartjuni_frankrijk.reproductiegetal_fr
    (1:18));
beteaseirs0 = (reproductieseirs0*(sigma+mu)*(gamma+mu))/sigma;

% Plot of reproductionnumber
plot(maartjuni_frankrijk.datums, maartjuni_frankrijk.
    reproductiegetal_fr);
xlim([datetime('01-03-2020', 'InputFormat', 'dd-MM-yyyy')
    datetime('01-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel("Data");
ylabel("Reproductiegetal");
hold on;
plot(beginlockdown, 0, 'ro', 'MarkerSize', 6, 'LineWidth', 2);
text(beginlockdown, 0, '17-03', 'VerticalAlignment', 'top', '
    HorizontalAlignment', 'center');
hold off;

save('tabelFR.mat', 'maartjuni_frankrijk'); % Save table to use
    in other file

% Plot of lockdownstrength in SIR-model
lockdown = 1-((maartjuni_frankrijk.reproductiegetal_fr(17:72)
    *0.1429)/beta0);
lockdowntabel_fr = table(maartjuni_frankrijk.datums(17:72),
    lockdown);
gemiddeldelockdown = mean(lockdown);
plot(lockdowntabel_fr.Var1, lockdowntabel_fr.lockdown);
xlabel("Data")
ylabel("Lockdownstrength")

% Plot of lockdown strength in SEIRS-model
lockdownseirs = 1-((maartjuni_frankrijk.reproductiegetal_fr
    (17:72)*(gamma+mu)*(sigma+mu))/(sigma*beta0));
lockdowntabel_fr = table(maartjuni_frankrijk.datums(17:72),
    lockdown, lockdownseirs);
gemiddeldelockdownseirs = mean(lockdowntabel_fr.lockdownseirs);
plot(lockdowntabel_fr.Var1, lockdowntabel_fr.lockdownseirs);
xlabel("Data")
ylabel("Lockdownstrength")
```

```
save('lockdowntabelFR.mat', 'lockdowntabel_fr'); % Save table to
   use in other file
```

I Appendix I

```
clear;
close all

% Read reproductionnumbers data
data = readtable("reproductiegetallen.csv");

% Select columns to use
land = data.location;
datum = data.date;
reproductiegetal = data.reproduction_rate;

% Make table
tabel = table(land, datum, reproductiegetal);
tabel.land = cellstr(tabel.land);
land = "Italy";
index = tabel.land == land;

% Create new column with specific country only
italie = tabel.land(index);
datums = tabel.datum(index);
reproductiegetal_it = tabel.reproductiegetal(index);
italietabel = table(italie, datums, reproductiegetal_it);

% Select data from 01-03-2020 until 01-06-2020
startdatum = datetime('2020-02-24');
einddatum4 = datetime('2020-06-01');
beginlockdown = datetime('2020-03-09');
maartjuni_italie = italietabel(italietabel.datums >= startdatum &
    italietabel.datums <= einddatum4, :);

plot(maartjuni_italie.datums, maartjuni_italie.
   reproductiegetal_it);

% Calculate beta0 and u(t)
gamma = 1/7;
sigma = 1/7;
mu = 1/(76*365);
```

```
reproductie0 = mean(maartjuni_italie.reproductiegetal_it(1:15));
beta0 = reproductie0*gamma;
maartjuni_italie.lockdownbetagetal = maartjuni_italie.
   reproductiegetal_it*gamma;

reproductieseirs0 = mean(maartjuni_italie.reproductiegetal_it
   (1:15));
beteaseirs0 = (reproductieseirs0*(sigma+mu)*(gamma+mu))/sigma;

% Plot of reproductionnumber
plot(maartjuni_italie.datums, maartjuni_italie.
   reproductiegetal_it);
xlim([datetime('24-02-2020', 'InputFormat', 'dd-MM-yyyy')
   datetime('01-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel("Data");
ylabel("Reproductiegetal");
hold on;
plot(beginlockdown, 0, 'ro', 'MarkerSize', 6, 'LineWidth', 2);
text(beginlockdown, 0, '09-03', 'VerticalAlignment', 'top', '
   HorizontalAlignment', 'center');
hold off;

save('tabelIT.mat', 'maartjuni_italie'); % Save table to use in
   other file

% Plot of lockdown strength for SIR-model
lockdown = 1-((maartjuni_italie.reproductiegetal_it(15:71)
   *0.1429)/beta0);
lockdowntabel_it = table(maartjuni_italie.datums(15:71), lockdown
   );
gemiddeldelockdown = mean(lockdowntabel_it.lockdown);
plot(lockdowntabel_it.Var1, lockdowntabel_it.lockdown);
xlabel("Data")
ylabel("Lockdownstrength")

% Plot of lockdown strength for SEIRS-model
alpha = 0;
lockdownseirs = 1-((maartjuni_italie.reproductiegetal_it(15:71)*(
   alpha+gamma+mu)*(sigma+mu))/(sigma*beta0));
lockdowntabel_it = table(maartjuni_italie.datums(15:71), lockdown
   , lockdownseirs);
gemiddeldelockdownseirs = mean(lockdowntabel_it.lockdownseirs);
```

```
plot(lockdowntabel_it.Var1, lockdowntabel_it.lockdownseirs);
xlabel("Data")
ylabel("Lockdownstrength")

save('lockdowntabelIT.mat', 'lockdowntabel_it');  % Save table to
    use in other file
```

J Appendix J

```
clear;
close all

% Read reproductionnumbers data
data = readtable("reproductiegetallen.csv");

% Select columns to use
land = data.location;
datum = data.date;
reproductiegetal = data.reproduction_rate;

% Make table
tabel = table(land, datum, reproductiegetal);
tabel.land = cellstr(tabel.land);
land = "Spain";
index = tabel.land == land;

% Create new column only with specific country
spanje = tabel.land(index);
datums = tabel.datum(index);
reproductiegetal_s = tabel.reproductiegetal(index);
spanjetabel = table(spanje, datums, reproductiegetal_s);

% Select data from 01-03-2020 until 01-06-2020
startdatum = datetime('2020-03-03');
einddatum4 = datetime('2020-06-21');
beginlockdown = datetime('2020-03-14');
maartjuni_spanje = spanjetabel(spanjetabel.datums >= startdatum &
    spanjetabel.datums <= einddatum4, :);

plot(maartjuni_spanje.datums, maartjuni_spanje.reproductiegetal_s
   ); % Curve of reproduction numbers of country

% Calculate beta0 and u(t)
```

```
gamma = 1/7;
sigma = 1/7;
mu = 1/(76*365);
reproductie0 = mean(maartjuni_spanje.reproductiegetal_s(1:12));
beta0 = reproductie0*gamma;
maartjuni_spanje.lockdownbetagetal = maartjuni_spanje.
   reproductiegetal_s*gamma;
%plot(maartjuni_italie.datums, maartjuni_italie.lockdownbetagetal
   );

reproductieseirs0 = mean(maartjuni_spanje.reproductiegetal_s
   (1:15));
beteaseirs0 = (reproductieseirs0*(sigma+mu)*(gamma+mu))/sigma;

% Plot of reproductionnumbers
plot(maartjuni_spanje.datums, maartjuni_spanje.reproductiegetal_s
   );
xlim([datetime('01-03-2020', 'InputFormat', 'dd-MM-yyyy')
   datetime('21-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel("Data");
ylabel("Reproductiegetal");
hold on;
plot(beginlockdown, 0, 'ro', 'MarkerSize', 6, 'LineWidth', 2);
text(beginlockdown, 0, '14-03', 'VerticalAlignment', 'top', '
   HorizontalAlignment', 'center');
hold off;

save('tabelSP.mat', 'maartjuni_spanje'); % Save table to use in
   other file

% Plot of lockdown strength for SIR-model
lockdown = 1-((maartjuni_spanje.reproductiegetal_s(12:111)
   *0.1429)/beta0);
lockdowntabel_s = table(maartjuni_spanje.datums(12:111), lockdown
   );
gemiddeldelockdown = mean(lockdowntabel_s.lockdown);
plot(lockdowntabel_s.Var1, lockdowntabel_s.lockdown);
xlabel("Data")
ylabel("Lockdownstrength")

% Plot of lockdown strength for SEIRS-model
lockdownseirs = 1-((maartjuni_spanje.reproductiegetal_s(12:111)*(
```

```
   gamma+mu)*(sigma+mu))/(sigma*beta0));
lockdowntabel_s = table(maartjuni_spanje.datums(12:111), lockdown
   , lockdownseirs);
gemiddeldelockdownseirs = mean(lockdowntabel_s.lockdownseirs);
plot(lockdowntabel_s.Var1, lockdowntabel_s.lockdownseirs);
xlabel("Data")
ylabel("Lockdownstrength")

save('lockdowntabelSP.mat', 'lockdowntabel_s'); % Save table to
   use in other file
```

K Appendix K

```
clear;
close all;

load("coronanederland_verbeterde_parameters.mat"); % Table from
   other file

% Set Parameters
Rt = 2;          % Transmission rate
P = 17395687;    % Population
I0 = 12;         % Initial infected population
tmax = 150;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
I(1) = I0;         % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;                    % Transmission rate (per day)
gamma = 1/7;                      % Recovery rate (per day)
lockdown_values = [0, 0.6118];  % Lockdown strength

beginlockdown = 15;  % Start lockdown
eindlockdown = 72;   % End lockdown
```

```
beginsimulatie = 1;  % Start day
eindsimulatie = 92;  % End day

figure;
hold on;

for k = 1:length(lockdown_values)
    lockdown = lockdown_values(k);

    % Step function
    u = zeros(1, nt);
    u(t >= 0 & t < 15) = 0;
    u(t >= 15 & t < 72) = lockdown;
    u(t >= 72) = 0;

    % Reset initial values for each lockdown value
    S(1) = P - I0;
    I(1) = I0;
    R(1) = 0;

    % Create SIR model
    for i = 2:nt
        dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;                %
            Change in S
        S(i) = S(i-1) + dS;                                      %
            Current S value
        if S(i) < 0; S(i) = 0; end

        dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i-1))*dt;    %
            Change in I
        I(i) = I(i-1)+dI;                                        %
            Current I value
        if I(i) < 0; I(i) = 0; end

        dR = (gamma*I(i-1))*dt;                                  %
            Change in R
        R(i) = R(i-1) + dR;                                      %
            Current R value
    end
    grootste1 = max(I(:));
    disp(grootste1)
    oppervlakte = trapz(t, I(:));
```

```
    begindag = 0;
    einddag = 92;
    beginlockdown = 15;
    eindlockdown = 72;
    simulatieeinde = einddag;

    % Plot of I graph
    plot(t, I, 'DisplayName', sprintf('Lockdown: %.2f', lockdown)
        );

end

% Add vertical lines
xline(12, '--');
xline(70, '--');
xline(91, 'LineWidth', 1);

hold off;

% Add xlabel, ylabel, and legend
xas = xlabel('Time (in days)');
xaspositie = get(xas, 'Position');
xasnieuwepositie = xaspositie - 0.1;
set(xas, 'Position', xasnieuwepositie)

ylabel ('Infected people (n)')
xticks([]);
    hold on;
    %text(begindag, 0, '01-03', 'VerticalAlignment', 'bottom', '
        HorizontalAlignment', 'center');
    text(beginlockdown, 0, '15-03: start lockdown', '
        VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
        ;
    text(eindlockdown, 0, '11-05: end lockdown', '
        VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
        ;
    text(simulatieeinde, 0, '01-06: period ending', '
        VerticalAlignment', 'top', 'HorizontalAlignment', 'left');
    hold off;

legend('u(t) = 0', 'u(t) = 0.61', 'Location', 'Northeast');
```

L Appendix L

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 17395687;    % Population
I0 = 12;         % Initial infected population
tmax = 150;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Define beta and gamma values
beta_values = 0:dt:1;
gamma_values = 0:dt:1;

% Define lockdown values
lockdown_values = 0.6118;

% Create figure
figure;

for l = 1:length(lockdown_values)
    % Initialize arrays for maximum I values
    max_I_values = zeros(length(gamma_values), length(beta_values
       ));

    % Loop over beta en gamma waarden
    for i = 1:length(beta_values)
        for j = 1:length(gamma_values)

            % Initialize values
            t=0:dt:tmax;      % Time values (in days)
            nt = length(t);   % Number of timesteps
            S = zeros(1, nt); % Susceptible values
            I = zeros(1, nt); % Infected values
            R = zeros(1, nt); % Recovered values
            S(1) = P - I0;    % Initial value of S
            I(1) = I0;        % Initial value of I
            R(1) = 0;         % Initial value of R

            % Calculations
```

```
        beta = beta_values(i);          % Transmission rate (
            per day)
        gamma = gamma_values(j);        % Recovery rate (per
            day)
        lockdown = lockdown_values(l);  % Lockdown strength

        % Create step function for u
        u = zeros(1, nt);
        u(t >= 0 & t < 15) = 0;
        u(t >= 15 & t < 72) = lockdown;
        u(t >= 72) = 0;

        % Create SIR model
        for k = 2:nt
            dS = (-beta*(1-u(k))*I(k-1)*S(k-1)/P)*dt;
                            % Change in S
            S(k) = S(k-1) + dS;
                                                % Current S
                value
            if S(k) < 0; S(k) = 0; end

            dI = (beta*(1-u(k))*I(k-1)*S(k-1)/P-gamma*I(k-1))
                *dt; % Change in I
            I(k) = I(k-1)+dI;
                                                  % Current S
                 value
            if I(k) < 0; I(k) = 0; end

            dR = (gamma*I(k-1))*dt;
                                            % Change in R
            R(k) = R(k-1) + dR;
                                                % Current R
                value
        end

        % Find maximum I value and save it
        max_I = max(I);
        max_I_values(j,i) = max_I;

    end

end
```

```
    % Create meshgrid for beta en gamma waarden
    [BETA, GAMMA] = meshgrid(beta_values, gamma_values);

    % Plot 3D surface
    subplot(1,2,l*2-1);
    surf(BETA, GAMMA, max_I_values);
    shading flat;
    colormap summer;
    xlabel('\beta');
    ylabel('\gamma');
    zlabel('Maximum I');
    title(['u(t) = ', num2str(lockdown_values(l))]);

    % Create contour plot
    subplot(1,2,l*2);
    contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
    hold on;
    j = plot(0.4137, 1/7, 'ro', 'MarkerSize', 5, 'LineWidth', 2);
    colormap summer;
    cb = colorbar;
    title(cb, 'I_{max}');
    xlabel('Transmission Rate (\beta)');
    ylabel('Recovery Rate (\gamma)');
    title(['u(t) = ', num2str(lockdown_values(l))]);

    % Create legend for the red circle
    legend(j, 'I_{max} = 4,998,200', 'Location', 'northwest');
     % Find and print the value of I_max for beta = 0.4137 and
        gamma = 1/7
    [~, idx] = min(abs(beta_values - 0.4137));
    [~, idy] = min(abs(gamma_values - 1/7));
    I_max = max_I_values(idy, idx);
    fprintf('I_max for beta = 0.4137 and gamma = 1/7: %.4f\n',
       I_max);
end

sgtitle('Lockdown btw t_1 = 15 and t_2 = 72', 'FontSize', 11, '
   FontWeight', 'bold');
```

M Appendix M

```
clear;
```

```
close all;

load("coronanederland_verbeterde_parameters.mat"); % Use table
    saved in other file

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
I0 = 1200;       % Initial infected population
tmax = 91;       % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
I(1) = I0;         % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;                    % Transmission rate (per day)
gamma = 1/7;                      % Recovery rate (per day)
lockdown_values = [0, 0.6118];    % Lockdown strength

beginlockdown = 15;  % Start lockdown
eindlockdown = 72;   % End lockdown
beginsimulatie = 1;  % Begindag
eindsimulatie = 92;  % Einddag

% Create figure
figure;
hold on;

for k = 1:length(lockdown_values)
    lockdown = lockdown_values(k);

    % Step function
    u = zeros(1, nt);
    u(t >= 0 & t < 15) = 0;
```

```
    u(t >= 15 & t < 72) = lockdown;
    u(t >= 72) = 0;

    % Reset initial values for each lockdown value
    S(1) = P - I0;
    I(1) = I0;
    R(1) = 0;

    % Create SIR model
    for i = 2:nt
        dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;                % Change in S
        S(i) = S(i-1) + dS;                                      % Current S value
        if S(i) < 0; S(i) = 0; end

        dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i-1))*dt;    % Change in I
        I(i) = I(i-1)+dI;                                        % Current I value
        if I(i) < 0; I(i) = 0; end

        dR = (gamma*I(i-1))*dt;                                  % Change in R
        R(i) = R(i-1) + dR;                                      % Current R value
    end
    grootste1 = max(I(:));
    disp(grootste1)
    oppervlakte = trapz(t, I(:));

    begindag = 0;
    einddag = 92;
    beginlockdown = 15;
    eindlockdown = 72;
    simulatieeinde = einddag;

    % Plot of I graph
    plot(t, I, 'DisplayName', sprintf('Lockdown: %.2f', lockdown)
       );

end
```

```
% Plot infection curve from other table
plot(t, I_nieuw);
xline(12, '--');
xline(70, '--');
xline(91, 'LineWidth', 1);

hold off;

% Add xlabel, ylabel, and legends
xas = xlabel('Time (in days)');
xaspositie = get(xas, 'Position');
xasnieuwepositie = xaspositie - 0.1;
set(xas, 'Position', xasnieuwepositie)

ylabel ('Infected people (n)')
xticks([]);
    hold on;
    text(beginlockdown, 0, '15-03: start lockdown', '
       VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
       ;
    text(eindlockdown, 0, '11-05: end lockdown', '
       VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
       ;
    text(simulatieeinde, 0, '01-06: period ending', '
       VerticalAlignment', 'top', 'HorizontalAlignment', 'left');
    hold off;

legend('Without time dependent lockdown', 'With time dependent
   lockdown', 'Location', 'North');
```

N Appendix N

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 200;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
```

```
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
E(1) = 1000;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.6;        % Transmission rate (per day)
gamma = 1/14;      % Recovery rate (per day)
mu = 1/(76*365);  % Death and birthrate (per day)
omega = 1/365;     % Rate of loss of immunity (per day)
sigma = 1/3;       % Latency period (in days)
alpha = 0.2;       % Infection-induced death ratio (per day)

% Create lockdown strengths
lockdowns = [0, 0.75];

% Create SEIRS model for each lockdown strength
figure; hold on;
for k = 1:length(lockdowns)
    % Reset initial values
    S(1) = P - I0;
    E(1) = 1000;
    I(1) = 0;
    R(1) = 0;

    % Set lockdown strength
    lockdown = lockdowns(k);

    % Create step function for u
    u = zeros(1, nt);
    u(t >= 0 & t < 50) = 0;
    u(t >= 50 & t < 70) = lockdown;
    u(t >= 70) = 0;

    % Create SEIRS model
    for i = 2:nt
```

```
        dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i-1)*S(i-1)/P)
           + omega*R(i-1))*dt;     % Change in S
        S(i) = S(i-1) + dS;

           % Current S value
        if S(i) < 0; S(i) = 0; end

        dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E(i-1) - mu*E
           (i-1))*dt;              % Change in E
        E(i) = E(i-1) + dE;

           % Current E value
        if E(i) < 0; E(i) = 0; end

        dI = (sigma*E(i-1) - gamma*I(i-1) - (mu + alpha)*I(i-1))*
           dt;                     % Change in I
        I(i) = I(i-1) + dI;

           % Current I value
        if I(i) < 0; I(i) = 0; end

        dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1))*dt;
                                      % Change in R
        R(i) = R(i-1) + dR;

           % Current R value
    end

    grootste = max(I(:));              % Maximum value of I
    disp(grootste)
    oppervlakte = trapz(t, I(:));      % Surface under I graph
    disp(oppervlakte)

    % Plot of SEIRS model
    plot(t, I)
end

hold off;

% Add xlabel, ylabel, and legends
legend('Lockdown = 0', 'Lockdown = 0.75');
legenda1 = legend('u(t) = 0', 'u(t) = 0.75');
```

```
set(legenda1, 'Position', [0.75, 0.78, 0.15, 0.05])
legendatekst1 = 'Lockdown start: 80';
legendatekst2 = 'Lockdown end: 100';
legendapositie = [0.72, 0.85, 0.18, 0.06]; %x,y,breedte,hoogte
legenda2 = annotation('textbox', legendapositie, 'String', {
   legendatekst1, legendatekst2}, 'FontSize', 9);
xlabel('Time (in days)');
ylabel ('Infected people (n)');
```

O Appendix O

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 200;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
E(1) = 1000;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.6;        % Transmission rate (per day)
gamma = 1/14;      % Recovery rate (per day)
mu = 1/(76*365);   % Death and birthrate (per day)
omega = 1/365;     % Rate of loss of immunity (per day)
sigma = 1/7;       % Latency period (in days)
alpha = 0.2;       % Infection-induced death ratio (per day)
lockdown = 0.75;   % Lockdown strength

startdagen = [70, 82, 92, 30]; % Starting days
```

```
einddagen = [90, 102, 112 50]; % Ending days

% Create figure
figure;
hold on;

% Create SEIRS model for each starting day and ending day
for k = 1:length(startdagen)
    startdag = startdagen(k);
    einddag = einddagen(k);

    % Reset initial values
    S(1) = P - I0;
    E(1) = 1000;
    I(1) = 0;
    R(1) = 0;

    % Create step function for u
    u = zeros(1, nt);
    u(t >= 0 & t < startdag) = 0;
    u(t >= startdag & t < einddag) = lockdown;
    u(t >= einddag) = 0;

    % Create SEIRS model
    for i = 2:nt
        dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i-1)*S(i-1)/P)
           + omega*R(i-1))*dt;     % Change in S
        S(i) = S(i-1) + dS;

           % Current S value
        if S(i) < 0; S(i) = 0; end

        dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E(i-1) - mu*E
           (i-1))*dt;              % Change in E
        E(i) = E(i-1) + dE;

           % Current E value
        if E(i) < 0; E(i) = 0; end

        dI = (sigma*E(i-1) - gamma*I(i-1) - (mu + alpha)*I(i-1))*
           dt;                     % Change in I
        I(i) = I(i-1) + dI;
```

```
            % Current S value
        if I(i) < 0; I(i) = 0; end

        dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1))*dt;
                                        % Change in R
        R(i) = R(i-1) + dR;

            % Current R value
    end
    % Create plot
    plot(t, I, 'DisplayName', sprintf('Lockdown start: %d,
       Lockdown end: %d', startdag, einddag));
end

% Add xlabel, ylabel, legends
hold off;
legend('Location', 'northwest');
xlabel('Time (in days)');
ylabel('Infected people (n)');
legendatekst1 = 'u(t) = 0.75';
legendapositie = [0.145, 0.73, 0.1, 0.04]; %x,y,breedte,hoogte
legenda2 = annotation('textbox', legendapositie, 'String', {
   legendatekst1}, 'FontSize', 9);
```

P Appendix P

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 200;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
```

```
R = zeros(1, nt); % Recovered values
S(1) = P - I0;    % Initial value of S
E(1) = 1000;      % Initial value of E
I(1) = 0;         % Initial value of I
R(1) = 0;         % Initial value of R

% Define parameter values
beta_values = 0:dt:1;
gamma_values = 0:dt:1;
startdag = 80;
einddag = 100;
lockdown = 0;
mu = 1/(76*365);  % Death and birthrate (per day)
omega = 1/365;    % Rate of loss of immunity (per day)
sigma = 1/7;      % Latency period (in days)
alpha = 0.2;      % Infection-induced death ratio (per day)

max_I_values = zeros(length(gamma_values), length(beta_values));
    % Store maximum I values

% Create SEIRS model for each combination of beta and gamma
for i = 1:length(beta_values)
    for j = 1:length(gamma_values)
        beta = beta_values(i);
        gamma = gamma_values(j);

        % Reset initial values
        S(1) = P - I0;
        E(1) = 1000;
        I(1) = 0;
        R(1) = 0;

        % Create step function for u
        u = zeros(1, nt);
        u(t >= 0 & t <  startdag) = 0;
        u(t >= startdag & t < einddag) = lockdown;
        u(t >= einddag) = 0;

        % Create SEIRS model
        for k = 2:nt
            dS = (mu*P - mu*S(k-1) - (beta*(1-u(k))*I(k-1)*S(k-1)
               /P) + omega*R(k-1))*dt;    % Change in S
```

```
            S(k) = S(k-1) + dS;

               % Current S value
            if S(k) < 0; S(k) = 0; end

            dE = (beta*(1-u(k))*I(k-1)*S(k-1)/P - sigma*E(k-1) -
               mu*E(k-1))*dt;              % Change in E
            E(k) = E(k-1) + dE;

               % Current E value
            if E(k) < 0; E(k) = 0; end

            dI = (sigma*E(k-1) - gamma*I(k-1) - (mu + alpha)*I(k
               -1))*dt;                    % Change in I
            I(k) = I(k-1) + dI;

               % Current S value
            if I(k) < 0; I(k) = 0; end

            dR = (gamma*I(k-1) - mu*R(k-1) - omega*R(k-1))*dt;
                                           % Change in R
            R(k) = R(k-1) + dR;

               % Current R value
        end

        % Store maximum I value
        max_I_values(j,i) = max(I);
    end
end

% Create meshgrid
[beta_mesh, gamma_mesh] = meshgrid(beta_values, gamma_values);

% Plot maximum I values with 4 surf plots and 4 contour plots
figure;
% Plot 3D surface 1
subplot(4,2,1);
surf(beta_mesh, gamma_mesh, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
```

```
ylabel('\gamma');
zlabel('Maximum I');
title('u(t) = 0');

% Create contour 1
subplot(4,2,2);
contourf(beta_mesh, gamma_mesh, max_I_values, 'LineColor', 'none'
   );
colormap summer;
cb = colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
title(cb, 'I_{max}');
title('u(t) = 0');

% Plot 3D surface 2
subplot(4,2,3);
surf(beta_mesh, gamma_mesh, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
zlim([0,1e6]);
xlim([0, 1])
ylim([0, 1]);
clim([0, 1e6]);
title('u(t) = 0');

% Create contour plot 2
subplot(4,2,4);
contourf(beta_mesh, gamma_mesh, max_I_values, 'LineColor', 'None'
   );
colormap summer;
colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
zlim([0,1e6]);
xlim([0,1])
ylim([0, 1]);
clim([0, 1e6]);
cb = colorbar('Ticks', 0:2e5:1e6);
```

```
title(cb, 'I_{max}');
title('u(t) = 0');

% Plot 3D surface 3
subplot(4,2,5);
surf(beta_mesh, gamma_mesh, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
zlim([0,5e5]);
xlim([0, 1])
ylim([0, 1]);
clim([0, 5e5]);
title('u(t) = 0');

% Create contour plot 4
subplot(4,2,6);
contourf(beta_mesh, gamma_mesh, max_I_values, 'LineColor', 'None'
   );
colormap summer;
colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
zlim([0,5e5]);
xlim([0,1])
ylim([0, 1]);
clim([0, 5e5]);
cb = colorbar('Ticks', 0:1e5:5e5);
title(cb, 'I_{max}');
title('u(t) = 0');

% Plot 3D surface 4
subplot(4,2,7);
surf(beta_mesh, gamma_mesh, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
zlim([0,20000]);
```

```
xlim([0, 1])
ylim([0, 1]);
clim([0, 20000]);
title('u(t) = 0');

% Create contour plot 4
subplot(4,2,8);
contourf(beta_mesh, gamma_mesh, max_I_values, 'LineColor', 'None'
   );
colormap summer;
colorbar;
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
zlim([0,20000]);
xlim([0,1])
ylim([0, 1]);
clim([0, 20000]);
cb = colorbar('Ticks', 0:5000:20000);
title(cb, 'I_{max}');
title('u(t) = 0');
sgtitle('Lockdown btw t_1 = 80 and t_2 = 100', 'FontSize', 11, '
   FontWeight', 'bold')
```

Q Appendix Q

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 200;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
```

```
E(1) = 1000;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Define parameter values
beta_values = 0:dt:1;
gamma_values = 0:dt:1;
lockdown_values = [0.35, 0.55, 0.75];
startdag = 80;
einddag = 100;
lockdown = 0.75;
mu = 1/(76*365);  % Death and birthrate (per day)
omega = 1/365;    % Rate of loss of immunity (per day)
sigma = 1/7;      % Latency period (in days)
alpha = 0.2;      % Infection-induced death ratio (per day)

% Create figure
figure;
for l = 1:length(lockdown_values)
    max_I_values = zeros(length(gamma_values), length(beta_values
       ));  % Store maximum I values

    % Create SEIRS model for each combination of beta and gamma
    for i = 1:length(beta_values)
        for j = 1:length(gamma_values)
            beta = beta_values(i);
            gamma = gamma_values(j);
            lockdown = lockdown_values(l);

            % Reset initial values
            S(1) = P - I0;
            E(1) = 1000;
            I(1) = 0;
            R(1) = 0;

            % Create step function for u
            u = zeros(1, nt);
            u(t >= 0 & t <  startdag) = 0;
            u(t >= startdag & t < einddag) = lockdown;
            u(t >= einddag) = 0;

            % Create SEIRS model
```

```
        for k = 2:nt
            dS = (mu*P - mu*S(k-1) - (beta*(1-u(k))*I(k-1)*S(
               k-1)/P) + omega*R(k-1))*dt;     % Change in S
            S(k) = S(k-1) + dS;

               % Current S value
            if S(k) < 0; S(k) = 0; end

            dE = (beta*(1-u(k))*I(k-1)*S(k-1)/P - sigma*E(k
               -1) - mu*E(k-1))*dt;              % Change in E
            E(k) = E(k-1) + dE;

               % Current E value
            if E(k) < 0; E(k) = 0; end

            dI = (sigma*E(k-1) - gamma*I(k-1) - (mu + alpha)*
               I(k-1))*dt;                       % Change in I
            I(k) = I(k-1) + dI;

               % Current S value
            if I(k) < 0; I(k) = 0; end

            dR = (gamma*I(k-1) - mu*R(k-1) - omega*R(k-1))*dt
               ;                                 % Change in R
            R(k) = R(k-1) + dR;

               % Current R value
        end

        % Store maximum I value
        max_I_values(j,i) = max(I);
    end
end

% Create meshgrid
[BETA, GAMMA] = meshgrid(beta_values, gamma_values);

% Plot 3D surface
subplot(3,2,l*2-1);
surf(BETA, GAMMA, max_I_values);
shading flat;
colormap summer;
```

```
    xlabel('\beta');
    ylabel('\gamma');
    zlabel('Maximum I');
    title(['u(t) = ', num2str(lockdown_values(l))]);

    % Create contour plot
    subplot(3,2,l*2);
    contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
    colormap summer;
    cb= colorbar;
    title(cb, 'I_{max}');
    xlabel('Transmission Rate (\beta)');
    ylabel('Recovery Rate (\gamma)');
    title(['u(t) = ', num2str(lockdown_values(l))]);
    sgtitle('Lockdown btw t_1 = 80 and t_2 = 100', 'FontSize',
       11, 'FontWeight', 'bold')
end
```

R Appendix R

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
I0 = 12;         % Initial infected population
tmax = 91;       % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;             % Time values (in days)
nt = length(t);          % Number of timesteps
S = zeros(1, nt);        % Susceptible values
E = zeros(1, nt);        % Incuation values
I_nieuw = zeros(1, nt); % Infected values
R = zeros(1, nt);        % Recovered values
S(1) = P - I0;           % Initial value of S
E(1) = 1200;             % Initial value of E
I_nieuw(1) = 0;          % Initial value of I
R(1) = 0;                % Initial value of R

% Calculations
```

```
beta = 1;           % Transmission rate (per day)
gamma = 1/7;        % Recovery rate (per day)
mu = 1/(76*365);    % Death and birthrate (per day)
omega = 1/365;      % Rate of loss of immunity (per day)
sigma = 1/5;        % Latency period (in days)
alpha = 1/(83*365); % Infection-induced death ratio (per day)

startdag = 15;
einddag = 72;

% REPRODUCTIONNUMBER SIR-model
load('tabelIT.mat', 'maartjuni_italie');
load('tabelFR.mat', 'maartjuni_frankrijk');
load('tabelSP.mat', 'maartjuni_spanje');

reproductiegetallen = table(maartjuni_frankrijk.datums(3:93),
   maartjuni_italie.reproductiegetal_it(9:99), maartjuni_frankrijk
   .reproductiegetal_fr(3:93), maartjuni_spanje.reproductiegetal_s
   (1:91));
reproductiegetallen.Properties.VariableNames{'Var1'} = 'Datums';
reproductiegetallen.Properties.VariableNames{'Var2'} = 'Italie';
reproductiegetallen.Properties.VariableNames{'Var3'} = 'Frankrijk
   ';
reproductiegetallen.Properties.VariableNames{'Var4'} = 'Spanje';
gemiddelde = mean(reproductiegetallen{:, {'Italie', 'Frankrijk',
   'Spanje'}}, 2);
reproductiegetallen.Gemiddelde = gemiddelde;
reproductiegetallen.Betagetal = reproductiegetallen.Gemiddelde*
   gamma;

R0_IT = mean(maartjuni_italie.reproductiegetal_it(1:15));
R0_FR = mean(maartjuni_frankrijk.reproductiegetal_fr(1:16));
R0_SP = mean(maartjuni_spanje.reproductiegetal_s(1:11));

beta0_IT = R0_IT*(sigma+mu)*(alpha+gamma+mu)/sigma;
beta0_FR = R0_FR*(sigma+mu)*(alpha+gamma+mu)/sigma;
beta0_SP = R0_SP*(sigma+mu)*(alpha+gamma+mu)/sigma;

beta0 = (beta0_IT + beta0_FR + beta0_SP)/3;

% PLOT OF REPRODUCTIONNUMBERS OF EACH COUNTRY AND MEAN
   REPRODUCTION LINE
```

```
figure;
plot(reproductiegetallen.Datums, reproductiegetallen.Italie);
hold on;
plot(reproductiegetallen.Datums, reproductiegetallen.Frankrijk);
plot(reproductiegetallen.Datums, reproductiegetallen.Spanje);
plot(reproductiegetallen.Datums, reproductiegetallen.Gemiddelde,
   'LineWidth', 2);
hold off;

xlim([datetime('01-03-2020', 'InputFormat', 'dd-MM-yyyy')
   datetime('01-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel("Dates");
ylabel("Reproduction numbers");
legend('Italy', 'France', 'Spain', 'Mean', 'location', 'northeast
   ');

saveas(gcf, 'reproductiegetallenITFRSP.png');

% LOCKDOWNS
lockdown_IT = 1-(reproductiegetallen.Italie(7:63)*(1/5 +
   1/(365*76))*(1/(83*365)+1/(365*76)+1/5)/(beta0*1/7));
lockdown_FR = 1-(reproductiegetallen.Frankrijk(15:70)*(1/5 +
   1/(365*76))*(1/(83*365)+1/(365*76)+1/5)/(beta0*1/7));
lockdown_SP = 1-(reproductiegetallen.Spanje(12:91)*(1/5 +
   1/(365*76))*(0+1/(365*76)+1/7)/(beta0*1/5));

u_IT = mean(lockdown_IT);
u_FR = mean(lockdown_FR);
u_SP = mean(lockdown_SP);
u0 = (u_IT + u_FR + u_SP)/3;

lockdown_IT2 = 1-(reproductiegetallen.Italie(13:70)*(1/5 +
   1/(365*76))*(1/(83*365)+1/(365*76)+1/7)/(beta0*1/5));
lockdown_FR2 = 1-(reproductiegetallen.Frankrijk(13:70)*(1/5 +
   1/(365*76))*(1/(83*365)+1/(365*76)+1/7)/(beta0*1/5));
lockdown_SP2 = 1-(reproductiegetallen.Spanje(13:70)*(1/5 +
   1/(365*76))*(1/(83*365)+1/(365*76)+1/7)/(beta0*1/5));
gemiddeldelockdown = 1-(reproductiegetallen.Gemiddelde(13:70)
   *(1/5 + 1/(365*76))*(1/(83*365)+1/(365*76)+1/7)/(beta0*1/5));

% PLOT OF LOCKDOWNS
figure;
```

```
plot(reproductiegetallen.Datums(13:70), lockdown_IT2);
hold on;
plot(reproductiegetallen.Datums(13:70), lockdown_FR2);
plot(reproductiegetallen.Datums(13:70), lockdown_SP2);
plot(reproductiegetallen.Datums(13:70), gemiddeldelockdown, '
   LineWidth', 2);
hold off;

xlim([datetime('01-03-2020', 'InputFormat', 'dd-MM-yyyy')
   datetime('01-06-2020', 'InputFormat', 'dd-MM-yyyy')]);
xlabel("Dates");
ylabel("Lockdownstrength");
legend('Italy', 'France', 'Spain', 'Mean', 'location', 'east');

saveas(gcf, 'lockdowngetallenITFRSP.png');

% Make column to use in SIR-model for beta and u(t)
goede = zeros(91,1);
goede(1:12) = beta0;
goede(13:70) = beta0*gemiddeldelockdown;
goede(71:91) = beta0;

reproductiegetallen.betametlockdown = goede;

% LOCKDOWNSTRENGTH SIR-model
load('lockdowntabelIT.mat', 'lockdowntabel_it');
load('lockdowntabelFR.mat', 'lockdowntabel_fr');
load('lockdowntabelSP.mat', 'lockdowntabel_s');

herhaling = reshape(repmat(reproductiegetallen.betametlockdown,
   1,100)', [], 1);

beginlockdown = 15;  % Start lockdown
eindlockdown = 72;   % End lockdown
beginsimulatie = 1;  % Beginday
eindsimulatie = 92;  % Endday

u(1) = herhaling(1);

% Create SEIRS model
for i = 2:nt
    u(i) = herhaling(i-1);
```

```
    dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I_nieuw(i-1)*S(i-1)/P
       ) + omega*R(i-1))*dt;     % Change in S
    S(i) = S(i-1) + dS;

       % Current S value
    if S(i) < 0; S(i) = 0; end

    dE = (beta*(1-u(i))*I_nieuw(i-1)*S(i-1)/P - sigma*E(i-1) - mu
       *E(i-1))*dt;              % Change in E
    E(i) = E(i-1) + dE;

       % Current E value
    if E(i) < 0; E(i) = 0; end

    dI = (sigma*E(i-1) - gamma*I_nieuw(i-1) - (mu + alpha)*
       I_nieuw(i-1))*dt;               % Change in I
    I_nieuw(i) = I_nieuw(i-1) + dI;
                                                              %
       Current S value
    if I_nieuw(i) < 0; I_nieuw(i) = 0; end

    dR = (gamma*I_nieuw(i-1) - mu*R(i-1) - omega*R(i-1))*dt;
                                      % Change in R
    R(i) = R(i-1) + dR;

       % Current R value
end

grootste = max(I_nieuw(:));              % Maximum value of I
disp(grootste)
oppervlakte = trapz(t, I_nieuw(:));      % Surface under I graph
disp(oppervlakte)

% Plot of I graph
plot(t,I_nieuw) % plot from other file
save('coronanederland_SEIRS_verbeterde_parameters.mat', 't', '
   I_nieuw');
xlabel('Time (in days)')
ylabel ('Infected people (n)')
```

S Appendix S

```
clear;
close all;

load("coronanederland_SEIRS_verbeterde_parameters.mat"); % Use
   table from other file

% Set Parameters
Rt = 2;          % Transmission rate
P = 17395687;    % Population
tmax = 250;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - E(1);  % Initial value of S
E(1) = 12;         % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;                    % Transmission rate (per day)
gamma = 1/7;                      % Recovery rate (per day)
mu = 1/(76*365);                  % Death and birthrate (per day)
omega = 1/365;                    % Rate of loss of immunity (per
   day)
sigma = 1/7;                      % Latency period (in days)
alpha = 0;                        % Infection-induced death ratio (
   per day)
lockdown_values = [0, 0.6118];  % Lockdown strength = 0.6118

beginlockdown = 15;  % Start lockdown
eindlockdown = 72;   % End lockdown
beginsimulatie = 1;  % Beginday
eindsimulatie = 92;  % Endday

startdag = 15;
einddag = 72;
```

```
figure;
hold on;

% Create SEIRS model
for k = 1:length(lockdown_values)
    lockdown = lockdown_values(k);

    u = zeros(1, nt);
    u(t >= 0 & t <  startdag) = 0;
    u(t >= startdag & t < einddag) = lockdown;
    u(t >= einddag) = 0;

    % Reset initial values
    S(1) = P - E(1);  % Initial value of S
    E(1) = 12;        % Initial value of E
    I(1) = 0;         % Initial value of I
    R(1) = 0;

    for i = 2:nt
        dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i-1)*S(i-1)/P)
           + omega*R(i-1))*dt;     % Change in S
        S(i) = S(i-1) + dS;

           % Current S value
        if S(i) < 0; S(i) = 0; end

        dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E(i-1) - mu*E
           (i-1))*dt;              % Change in E
        E(i) = E(i-1) + dE;

           % Current E value
        if E(i) < 0; E(i) = 0; end

        dI = (sigma*E(i-1) - gamma*I(i-1) - (mu + alpha)*I(i-1))*
           dt;                     % Change in I
        I(i) = I(i-1) + dI;

           % Current S value
        if I(i) < 0; I(i) = 0; end

        dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1))*dt;
```

```
                                        % Change in R
        R(i) = R(i-1) + dR;

            % Current R value
    end

    grootste = max(I(:));              % Maximum value of I
    disp(grootste)
    oppervlakte = trapz(t, I(:));      % Surface under I graph
    disp(oppervlakte)

    begindag = 0;
    einddag = 92;
    beginlockdown = 15;
    eindlockdown = 72;
    simulatieeinde = einddag;

    % Plot of I graph
    plot(t, I, 'DisplayName', sprintf('Lockdown: %.2f', lockdown)
       );
end

% Create vertical lines
xline(12, '--');
xline(70, '--');
xline(91, 'LineWidth', 1);

hold off;

% Add xlabel, ylabel, and legends
xas = xlabel('Time (in days)');
xaspositie = get(xas, 'Position');
xasnieuwepositie = xaspositie - 0.1;
set(xas, 'Position', xasnieuwepositie)

ylabel ('Infected people (n)')
xticks([]);
    hold on;
    text(beginlockdown, 0, '15-03: start lockdown', '
       VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
       ;
    text(eindlockdown, 0, '11-05: end lockdown', '
```

```
        VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
        ;
    text(simulatieeinde, 0, '01-06: period ending', '
        VerticalAlignment', 'top', 'HorizontalAlignment', 'left');
    hold off;

legend('u(t)=0', 'u(t)=0.61', 'Location', 'Northwest');
```

T Appendix T

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 17395687;    % Population
I0 = 12;         % Initial infected population
tmax = 250;      % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
E(1) = 12;         % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Define parameter values
beta_values = 0:dt:1;
gamma_values = 0:dt:1;
lockdown_values = 0.6118;
startdag = 15;
einddag = 72;
mu = 1/(76*365);  % Death and birthrate (per day)
omega = 1/365;    % Rate of loss of immunity (per day)
sigma = 1/7;      % Latency period (in days)
alpha = 0;        % Infection-induced death ratio (per day)
```

```
% Create figure
figure;
for l = 1:length(lockdown_values)
    max_I_values = zeros(length(gamma_values), length(beta_values
       ));  % Store maximum I values

    % Create SEIRS model for each combination of beta and gamma
    for i = 1:length(beta_values)
        for j = 1:length(gamma_values)
            beta = beta_values(i);
            gamma = gamma_values(j);
            lockdown = lockdown_values(l);

            % Reset initial values
            S(1) = P - I0;
            E(1) = 1000;
            I(1) = 0;
            R(1) = 0;

            % Create step function for u
            u = zeros(1, nt);
            u(t >= 0 & t <  startdag) = 0;
            u(t >= startdag & t < einddag) = lockdown;
            u(t >= einddag) = 0;

            % Create SEIRS model
            for k = 2:nt
                dS = (mu*P - mu*S(k-1) - (beta*(1-u(k))*I(k-1)*S(
                   k-1)/P) + omega*R(k-1))*dt;    % Change in S
                S(k) = S(k-1) + dS;

                   % Current S value
                if S(k) < 0; S(k) = 0; end

                dE = (beta*(1-u(k))*I(k-1)*S(k-1)/P - sigma*E(k
                   -1) - mu*E(k-1))*dt;             % Change in E
                E(k) = E(k-1) + dE;

                   % Current E value
                if E(k) < 0; E(k) = 0; end

                dI = (sigma*E(k-1) - gamma*I(k-1) - (mu + alpha)*
```

```
            I(k-1))*dt;                          % Change in I
        I(k) = I(k-1) + dI;

            % Current S value
        if I(k) < 0; I(k) = 0; end

        dR = (gamma*I(k-1) - mu*R(k-1) - omega*R(k-1))*dt
            ;                                    % Change in R
        R(k) = R(k-1) + dR;

            % Current R value
    end

    % Store maximum I value
    max_I_values(j,i) = max(I);
  end
end
% Create meshgrid
[BETA, GAMMA] = meshgrid(beta_values, gamma_values);

% Plot 3D surface
subplot(3,2,l*2-1);
surf(BETA, GAMMA, max_I_values);
shading flat;
colormap summer;
xlabel('\beta');
ylabel('\gamma');
zlabel('Maximum I');
title(['u(t) = ', num2str(lockdown_values(l))]);

% Create contour plot
subplot(3,2,l*2);
contourf(BETA, GAMMA, max_I_values, 'LineColor', 'None');
hold on;
j = plot(0.4137, 1/7, 'ro', 'MarkerSize', 5, 'LineWidth', 2);
colormap summer;
cb= colorbar;
title(cb, 'I_{max}');
xlabel('Transmission Rate (\beta)');
ylabel('Recovery Rate (\gamma)');
title(['u(t) = ', num2str(lockdown_values(l))]);
sgtitle('Lockdown btw t_1 = 15 and t_2 = 72', 'FontSize', 11,
```

```
        'FontWeight', 'bold')

    % Create legend for the red circle
    legend(j, 'I_{max} = 2,487,700', 'Location', 'northwest');
    [~, idx] = min(abs(beta_values - 0.4137));
    [~, idy] = min(abs(gamma_values - 1/7));
    I_max = max_I_values(idy, idx);
    fprintf('I_max for beta = 0.4137 and gamma = 1/7: %.4f\n',
       I_max);
end
```

U Appendix U

```
clear;
close all;

load("coronanederland_SEIRS_verbeterde_parameters.mat"); % Use
   table from other file

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
tmax = 91;       % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - E(1);  % Initial value of S
E(1) = 1200;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;                  % Transmission rate (per day)
gamma = 1/7;                    % Recovery rate (per day)
mu = 1/(76*365);                % Death and birthrate (per day)
omega = 1/365;                  % Rate of loss of immunity (per
   day)
```

```
sigma = 1/5;                    % Latency period (in days)
alpha = 1/(83*365);             % Infection-induced death ratio (
   per day)
lockdown_values = [0, 0.6118];  % Lockdown strength = 0.6118

beginlockdown = 15;  % Start lockdown
eindlockdown = 72;   % End lockdown
beginsimulatie = 1;  % Beginday
eindsimulatie = 92;  % Endday

startdag = 15;
einddag = 72;

figure;
hold on;

% Create SEIRS model
for k = 1:length(lockdown_values)
    lockdown = lockdown_values(k);

    u = zeros(1, nt);
    u(t >= 0 & t <  startdag) = 0;
    u(t >= startdag & t < einddag) = lockdown;
    u(t >= einddag) = 0;

    % Reset initial values
    S(1) = P - E(1);    % Initial value of S
    E(1) = 1200;        % Initial value of E
    I(1) = 0;           % Initial value of I
    R(1) = 0;

    for i = 2:nt
        dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i-1)*S(i-1)/P)
           + omega*R(i-1))*dt;     % Change in S
        S(i) = S(i-1) + dS;

           % Current S value
        if S(i) < 0; S(i) = 0; end

        dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E(i-1) - mu*E
           (i-1))*dt;              % Change in E
        E(i) = E(i-1) + dE;
```

```
        % Current E value
    if E(i) < 0; E(i) = 0; end

    dI = (sigma*E(i-1) - gamma*I(i-1) - (mu + alpha)*I(i-1))*
        dt;                     % Change in I
    I(i) = I(i-1) + dI;

        % Current S value
    if I(i) < 0; I(i) = 0; end

    dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1))*dt;
                                    % Change in R
    R(i) = R(i-1) + dR;

        % Current R value
end

grootste = max(I(:));               % Maximum value of I
disp(grootste)
oppervlakte = trapz(t, I(:));       % Surface under I graph
disp(oppervlakte)

begindag = 0;
einddag = 92;
beginlockdown = 15;
eindlockdown = 72;
simulatieeinde = einddag;

 % Plot of I graph
plot(t, I, 'DisplayName', sprintf('Lockdown: %.2f', lockdown)
    );
end

plot(t, I_nieuw); % Plot from other file
% Add vertical lines
xline(12, '--');
xline(70, '--');
xline(91, 'LineWidth', 1);

hold off;
```

```
% Add xlabel, ylabel, and legend
xas = xlabel('Time (in days)');
xaspositie = get(xas, 'Position');
xasnieuwepositie = xaspositie - 0.1;
set(xas, 'Position', xasnieuwepositie)
ylabel ('Infected people (n)')
xticks([]);
    hold on;
    %text(begindag, 0, '01-03', 'VerticalAlignment', 'bottom', '
       HorizontalAlignment', 'center');
    text(beginlockdown, 0, '15-03: start lockdown', '
       VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
       ;
    text(eindlockdown, 0, '11-05: end lockdown', '
       VerticalAlignment', 'top', 'HorizontalAlignment', 'center')
       ;
    text(simulatieeinde, 0, '01-06: period ending', '
       VerticalAlignment', 'top', 'HorizontalAlignment', 'left');
    hold off;

legend('Without time dependent lockdown', 'u(t) = With time
   dependent lockdown', 'Location', 'North');
```

V Appendix V

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
I0 = 1200;       % Initial infected population
tmax = 91;       % Length of simulation (in days)
dt = 0.1;        % Time step (in days)

% Initialize values
t = 0:dt:tmax;     % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
I(1) = I0;         % Initial value of I
```

```
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;       % Transmission rate (per day)
gamma = 1/7;         % Recovery rate (per day)
lockdown = 0.6118;   % Lockdown strength

t1 = 0:dt:tmax;   % Starting day of lockdown
t2 = 0:dt:tmax;   % Ending day of lockdown

aantal = ((tmax/dt)*((tmax/dt)+1))/2;
resultaten = zeros(aantal,3);

Z = zeros(length(t1), length(t2));

% Create SIR model with given cost function
l=1;
for j = 1:nt
    for k = 1:nt
        if t1(j) < t2(k)
            u(t >= 0 & t < t1(j)) = 0;
            u(t >= t1(j) & t < t2(k)) = lockdown;
            u(t >= t2(k)) = 0;

            resultaten(l,1) = t1(j);
            resultaten(l,2) = t2(k);

            for i = 2:nt
                dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;
                                % Change in S
                S(i) = S(i-1) + dS;
                                                % Current S
                    value
                if S(i) < 0; S(i) = 0; end

                dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i-1))
                   *dt;   % Change in I
                I(i) = I(i-1)+dI;
                                                % Current
                    I value
                if I(i) < 0; I(i) = 0; end
```

```
                dR = (gamma*I(i-1))*dt;
                                                % Change in R
                R(i) = R(i-1) + dR;
                                                % Current R
                    value

                %Z(j,k) = I(i);
            end
            grootste = max(I(:));               % Maximum value of
                I
            resultaten(l,3) = grootste;
            Z(j,k) = resultaten(l,3);
            oppervlakte = trapz(t, I(:));       % Surface under I
               curve
            resultaten(l,4) = oppervlakte;
            l=l+1;
        end
    end
end

[X,Y] = meshgrid(0:dt:tmax);

% Limit X and Y to the part for which t1 <= t2
X = triu(X);
Y = triu(Y);

% Limit Z to the part for which t1 <= t2
Z = triu(Z);

% Create three-dimensional and contour plot
figure;

% Surf plot
subplot(1, 2, 1);
surf(X, Y, Z.*(X>Y));
shading flat;
colormap summer;
xlabel('t_2');
ylabel('t_1');
zlabel('Maximum I');
annotation('textbox', [0.1, 0.8, 0.1, 0.1], 'String', 'u(t)
   =0.6118', 'FitBoxToText', 'on');
```

```
% Contour plot
subplot(1, 2, 2);
contourf(X, Y, Z.*(X>Y), 'LineColor', 'None');
hold on;
j = plot(72, 15, 'ro', 'MarkerSize', 5, 'LineWidth', 2);
k = plot(72, 21, 'bo', 'MarkerSize', 5, 'LineWidth', 2);
l = plot(76.5, 19.5, 'go', 'MarkerSize', 5, 'LineWidth', 2);
colormap summer;
cb= colorbar;
title(cb, 'I_{max}');
xlabel('End lockdown (t_2)');
ylabel('Start lockdown (t_1)');

% Create a combined legend
combined_legend = legend([j, k, l], {'I_{max} = 609020', 'I_{max}
    = 282210', 'I_{max} = 204360'}, 'Location', 'northwest');

% Adjust the position of the combined legend
legend_pos = get(combined_legend, 'Position');
legend_pos(2) = legend_pos(2) - 0.05;
set(combined_legend, 'Position', legend_pos);
```

W Appendix W

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
I0 = 1200;       % Initial infected population
tmax = 91;       % Length of simulation (in days)
dt = 0.01;       % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
I(1) = I0;         % Initial value of I
```

```
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;       % Transmission rate (per day)
gamma = 1/7;         % Recovery rate (per day)
lockdown = 0.6118;   % Lockdown strength
tijdsverschil = 57; % Duration

% Create SIR model
t1 = 0:dt:tmax;
t2 = t1 + tijdsverschil;

aantal = 1;
resultaten = zeros(aantal,3);

Z = zeros(length(t1),length(t1));

l=1;
for j = 1:nt
    if (t1(j) < t2(j))

        % Create stepfunction
        u(t >= 0 & t < t1(j)) = 0;
        u(t >= t1(j) & t < t2(j)) = lockdown;
        u(t >= t2(j)) = 0;

        resultaten(l,1) = t1(j);
        resultaten(l,2) = t2(j);

        for i = 2:nt
            dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;
                              % Change in S
            S(i) = S(i-1) + dS;
                                                  % Current S
               value
            if S(i) < 0; S(i) = 0; end

            dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i-1))*dt;
                  % Change in I
            I(i) = I(i-1)+dI;
                                                % Current I
               value
```

```
            if I(i) < 0; I(i) = 0; end

            dR = (gamma*I(i-1))*dt;
                                                % Change in R
            R(i) = R(i-1) + dR;
                                                    % Current R
               value

            Z(j,j) = I(i);
        end
        grootste = max(I(:));
        resultaten(l,3) = grootste;
        oppervlakte = trapz(t, I(:));
        resultaten(l,4) = oppervlakte;
        l=l+1;
    end
end

% Plot of minimum I_max for different starting days
opp2 = trapz(resultaten(:,1), resultaten(:,4));

plot(resultaten(:,1), resultaten(:,3))
hold on;
j = plot(15, resultaten(15/dt+1,3), 'ro', 'MarkerSize', 5, '
   LineWidth', 2);
k = plot(19.5, resultaten(19.5/dt+1,3), 'go', 'MarkerSize', 5, '
   LineWidth', 2);
hold off;
xlim([0 tmax])
xlabel('Time (in days)')
ylabel ('Maximum I')
annotation('textbox', [0.6, 0.3, 0.1, 0.1], 'String', 'u(t)
   =0.6118; duration = 57 days', 'FitBoxToText', 'on');
combined_legend = legend([j, k], {'I_{max} = 609020', 'I_{max} =
   204360'}, 'Location', 'east');
legend_pos = get(combined_legend, 'Position');
legend_pos(2) = legend_pos(2) - 0.05; % Adjust the vertical
   position as needed
set(combined_legend, 'Position', legend_pos);

laagste = min(resultaten(:,3));
[X,Y] = find(Z == laagste);
```

```
% Find row in which value of column 4 is equal to laagste
rij = find(resultaten(:,3) == laagste, 1);

% Time of day t1 that belongs to this row
startdag = resultaten(rij,1);
```

X Appendix X

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
I0 = 1000;       % Initial infected population
tmax = 91;       % Length of simulation (in days)
dt = 0.1;        % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
E(1) = 1200;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;       % Transmission rate (per day)
gamma = 1/7;         % Recovery rate (per day)
mu = 1/(76*365);     % Death and birthrate (per day)
omega = 1/365;       % Rate of loss of immunity (per day)
sigma = 1/5;         % Latency period (in days)
alpha = 1/(83*365); % Infection-induced death ratio (per day)
lockdown = 0.6118;  % Lockdown strength

t1 = 0:dt:tmax;  % Starting day of lockdown
t2 = 0:dt:tmax;  % Ending day of lockdown
```

```
aantal = ((tmax/dt)*((tmax/dt)+1))/2;
resultaten = zeros(aantal,3);

Z = zeros(length(t1), length(t2));

% Create SEIRS model with given lockdownstrength
l=1;
for j = 1:nt
    for k = 1:nt
        if t1(j) < t2(k)
            u(t >= 0 & t < t1(j)) = 0;
            u(t >= t1(j) & t < t2(k)) = lockdown;
            u(t >= t2(k)) = 0;

            resultaten(l,1) = t1(j);
            resultaten(l,2) = t2(k);

            for i = 2:nt
                dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i-1)*S(
                   i-1)/P) + omega*R(i-1))*dt;    % Change in S
                S(i) = S(i-1) + dS;

                   % Current S value
                if S(i) < 0; S(i) = 0; end

                dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E(i
                   -1) - mu*E(i-1))*dt;             % Change in E
                E(i) = E(i-1) + dE;

                   % Current E value
                if E(i) < 0; E(i) = 0; end

                dI = (sigma*E(i-1) - gamma*I(i-1) - (mu + alpha)*
                   I(i-1))*dt;                      % Change in I
                I(i) = I(i-1) + dI;

                   % Current S value
                if I(i) < 0; I(i) = 0; end

                dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1))*dt
                   ;                                % Change in R
                R(i) = R(i-1) + dR;
```

```
                % Current R value
            end

            grootste = max(I(:));              % Maximum value of
                I
            Z(j,k) = grootste(:);
            resultaten(l,3) = grootste;
            oppervlakte = trapz(t, I(:));      % Surface under I
               curve
            resultaten(l,4) = oppervlakte;
            l=l+1;
        end
    end
end

[X,Y] = meshgrid(0:dt:tmax);

% Limit X and Y to the part for which t1 <= t2
X = triu(X);
Y = triu(Y);

% Limit Z to the part for which t1 <= t2
Z = triu(Z);

% Create a new figure
figure;

% Surf plot
subplot(1, 2, 1);
surf(X, Y, Z.*(X>Y));
shading flat;
colormap summer;
xlabel('t_2');
ylabel('t_1');
zlabel('Maximum I');
annotation('textbox', [0.1, 0.8, 0.1, 0.1], 'String', 'u(t)
   =0.6118', 'FitBoxToText', 'on');

% Contour plot
subplot(1, 2, 2);
contourf(X, Y, Z.*(X>Y), 'LineColor', 'None');
```

```
hold on;
j = plot(72, 15, 'ro', 'MarkerSize', 5, 'LineWidth', 2);
colormap summer;
cb= colorbar;
title(cb, 'I_{max}');
xlabel('End lockdown (t_2)');
ylabel('Start lockdown (t_1)');
combined_legend = legend(j, {'I_{max} = 552,630'}, 'Location', '
   northwest');
legend_pos = get(combined_legend, 'Position');
legend_pos(2) = legend_pos(2) - 0.05;
set(combined_legend, 'Position', legend_pos);

kleinste = min(resultaten(:,3));
[X,Y] = find(Z == kleinste);
```

Y Appendix Y

```
clear;
close all;

% Set Parameters
Rt = 2;          % Transmission rate
P = 3400000;     % Population
I0 = 1000;       % Initial infected population
tmax = 91;       % Length of simulation (in days)
dt = 0.1;        % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
E(1) = 1200;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;       % Transmission rate (per day)
gamma = 1/7;         % Recovery rate (per day)
```

```
mu = 1/(76*365);     % Death and birthrate (per day)
omega = 1/365;       % Rate of loss of immunity (per day)
sigma = 1/5;         % Latency period (in days)
alpha = 1/(83*365); % Infection-induced death ratio (per day)
lockdown = 0.6118;  % Lockdown strength
tijdsverschil = 57; % Duration

% Create SIR model
t1 = 0:dt:tmax;
t2 = t1 + tijdsverschil;

aantal = 1;
resultaten = zeros(aantal,3);

Z = zeros(length(t1),length(t1));

l=1;
for j = 1:nt
    if (t1(j) < t2(j))

        % Create stepfunction
        u(t >= 0 & t < t1(j)) = 0;
        u(t >= t1(j) & t < t2(j)) = lockdown;
        u(t >= t2(j)) = 0;

        resultaten(l,1) = t1(j);
        resultaten(l,2) = t2(j);

        for i = 2:nt
                dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i-1)*S(
                   i-1)/P) + omega*R(i-1))*dt;    % Change in S
                S(i) = S(i-1) + dS;

                   % Current S value
                if S(i) < 0; S(i) = 0; end

                dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E(i
                   -1) - mu*E(i-1))*dt;             % Change in E
                E(i) = E(i-1) + dE;

                   % Current E value
                if E(i) < 0; E(i) = 0; end
```

```
                dI = (sigma*E(i-1) - gamma*I(i-1) - (mu + alpha)*
                    I(i-1))*dt;                         % Change in I
                I(i) = I(i-1) + dI;

                    % Current S value
                if I(i) < 0; I(i) = 0; end

                dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1))*dt
                    ;                                   % Change in R
                R(i) = R(i-1) + dR;

                    % Current R value
        Z(j,j) = I(i);
        end
        grootste = max(I(:));                % Maximum value of I
        resultaten(l,3) = grootste;
        oppervlakte = trapz(t, I(:));        % Surface under I
            graph
        resultaten(l,4) = oppervlakte;
        l=l+1;
    end
end

% Plot of minimum I_max for different starting days
opp2 = trapz(resultaten(:,1), resultaten(:,4));

plot(resultaten(:,1), resultaten(:,3))
hold on;
j = plot(15, resultaten(15/dt+1,3), 'ro', 'MarkerSize', 5, '
    LineWidth', 2);
hold off;
xlim([0 tmax])
xlabel('Time (in days)')
ylabel ('Maximum I')
annotation('textbox', [0.145, 0.7, 0.1, 0.1], 'String', 'u(t)
    =0.6118; duration = 57 days', 'FitBoxToText', 'on');
combined_legend = legend(j, {'I_{max} = 552,630'}, 'Location', '
    northwest');
legend_pos = get(combined_legend, 'Position');
legend_pos(2) = legend_pos(2) - 0.05;
set(combined_legend, 'Position', legend_pos);
```

```
laagste = min(resultaten(:,3));
[X,Y] = find(Z == laagste);

% Find row in which value of column 4 is equal to laagste
rij = find(resultaten(:,3) == laagste, 1);

% Time of day t1 that belongs to this row
startdag = resultaten(rij,1);
```

Z Appendix Z

```
clear;
close all;

beta_values = [0.3, 0.5, 0.7, 0.9]; % Values for beta

% Create figure
figure;
hold on;

for beta = beta_values
    x = [0.75, 20];      %x(1) = lockdown, x(2) = duration

    options = optimoptions('gamultiobj','PopulationSize',60,...
              'ParetoFraction',0.7);
    lb = [0 0];
    ub = [1 100];

    [solution, ObjectiveValue] = gamultiobj(@(x) eindfunctie(x,
       beta), 2,...
                                [],[],[],[],lb,ub,options);

    plot(ObjectiveValue(:,1), ObjectiveValue(:,2), 'o');
end

% Add xlabel, ylabel, and legend
xlabel('I_{max}');
ylabel('u_{tot}');
legend(string(beta_values), 'Location', 'northeast');

hold off;
```

```
% Create function
function laagste = mijnfunctie1(x, beta)

    % Set Parameters
    Rt = 2;           % Transmission rate
    P = 10000000;     % Population
    I0 = 1000;        % Initial infected population
    tmax = 100;       % Length of simulation (in days)
    dt = 2;           % Time step (in days)

    % Initialize values
    t=0:dt:tmax;       % Time values (in days)
    nt = length(t);    % Number of timesteps
    S = zeros(1, nt); % Susceptible values
    I = zeros(1, nt); % Infected values
    R = zeros(1, nt); % Recovered values
    S(1) = P - I0;     % Initial value of S
    I(1) = I0;         % Initial value of I
    R(1) = 0;          % Initial value of R

    % Calculations
    gamma = 0.2;       % Recovery rate (per day)

    t1 = 0:dt:tmax;
    t2 = t1 + x(2);

    aantal = 1;
    resultaten = zeros(aantal,3);

    Z = zeros(length(t1),length(t1));

    l=1;
    for j = 1:nt
        if (t1(j) < t2(j))
            u(t >= 0 & t < t1(j)) = 0;
            u(t >= t1(j) & t < t2(j)) = x(1);
            u(t >= t2(j)) = 0;

            resultaten(l,1) = t1(j);
            resultaten(l,2) = t2(j);

            for i = 2:nt
```

```
                dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;
                                % Change in S
                S(i) = S(i-1) + dS;
                                                % Current S
                    value
                if S(i) < 0; S(i) = 0; end

                dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i-1))
                   *dt;   % Change in I
                I(i) = I(i-1)+dI;
                                                % Current
                    I value
                if I(i) < 0; I(i) = 0; end

                dR = (gamma*I(i-1))*dt;
                                            % Change in R
                R(i) = R(i-1) + dR;
                                                % Current R
                    value

                Z(j,j) = I(i);
            end
            grootste = max(I(:));
            resultaten(l,3) = grootste;
            oppervlakte = trapz(t, I(:));
            resultaten(l,4) = oppervlakte;
            l=l+1;
        end
    end

    % Find minimum value of I_max
    laagste = min(resultaten(:,4));
    [X,Y] = find(Z == laagste);

    % Find row in which value of column 4 is equal to laagste
    rij = find(resultaten(:,4) == laagste, 1);

    % Starting day t1 that belongs to this row
    startdag = resultaten(rij,1);
end

% Create another function
```

```
function w = eindfunctie(x, beta)
    laagste = mijnfunctie1(x, beta);
    w(1) = laagste;
    w(2) = x(1)*x(2); %x(1) = lockdown, x(2) = duration
end
```

7 Appendix AA

```
clear;
close all;

beta_values = [0.4, 0.6, 0.8]; % Beta values

% Create figure
figure;
hold on;

for beta = beta_values
    x = [0.75, 20];      %x(1) = lockdown, x(2) = duration

    options = optimoptions('gamultiobj','PopulationSize',60,...
              'ParetoFraction',0.7);
    lb = [0 0];
    ub = [1 150];

    [solution, ObjectiveValue] = gamultiobj(@(x) eindfunctie(x,
       beta), 2,...
                                [],[],[],[],lb,ub,options);

    plot(ObjectiveValue(:,1), ObjectiveValue(:,2), 'o');
end

% Add xlabel, ylabel, and legend
xlabel('I_{max}');
ylabel('u_{tot}');
legend(string(beta_values), 'Location', 'northeast');

hold off;

% Create function
function laagste = mijnfunctie1(x, beta)

    % Set Parameters
```

```
Rt = 2;          % Transmission rate
P = 10000000;    % Population
I0 = 1000;       % Initial infected population
tmax = 150;      % Length of simulation (in days)
dt = 2;          % Time step (in days)

% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
E = zeros(1, nt); % Incuation values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
E(1) = 1000;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
gamma = 1/14;      % Recovery rate (per day)
mu = 1/(76*365);   % Death and birthrate (per day)
omega = 1/365;     % Rate of loss of immunity (per day)
sigma = 1/7;       % Latency period (in days)
alpha = 0.2;       % Infection-induced death ratio (per day)
lockdown = 0.75;   % Lockdown strength

t1 = 0:dt:tmax;
t2 = t1 + x(2);

aantal = 1;
resultaten = zeros(aantal,3);

Z = zeros(length(t1),length(t1));

l=1;
for j = 1:nt
    if (t1(j) < t2(j))
        u(t >= 0 & t < t1(j)) = 0;
        u(t >= t1(j) & t < t2(j)) = x(1);
        u(t >= t2(j)) = 0;

        resultaten(l,1) = t1(j);
```

```
        resultaten(l,2) = t2(j);

        for i = 2:nt
            dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i-1)*S(
               i-1)/P) + omega*R(i-1))*dt;    % Change in S
            S(i) = S(i-1) + dS;

               % Current S value
            if S(i) < 0; S(i) = 0; end

            dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E(i
               -1) - mu*E(i-1))*dt;             % Change in E
            E(i) = E(i-1) + dE;

               % Current E value
            if E(i) < 0; E(i) = 0; end

            dI = (sigma*E(i-1) - gamma*I(i-1) - (mu + alpha)*
               I(i-1))*dt;                      % Change in I
            I(i) = I(i-1) + dI;

               % Current S value
            if I(i) < 0; I(i) = 0; end

            dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1))*dt
               ;                                % Change in R
            R(i) = R(i-1) + dR;

               % Current R value
        Z(j,j) = I(i);
        end

        grootste = max(I(:));
        resultaten(l,3) = grootste;
        oppervlakte = trapz(t, I(:));
        resultaten(l,4) = oppervlakte;
        l=l+1;
    end
end

% Find minimum value of I_max
laagste = min(resultaten(:,4));
```

```
    [X,Y] = find(Z == laagste);

    % Find row in which value of column 4 is equal to laagste
    rij = find(resultaten(:,4) == laagste, 1);

    % Starting day t1 that belongs to this row
    startdag = resultaten(rij,1);
end

% Create another function
function w = eindfunctie(x, beta)
    laagste = mijnfunctie1(x, beta);
    w(1) = laagste;
    w(2) = x(1)*x(2); %x(1) = lockdown, x(2) = duration
end
```

8 Appendix AB

```
clear;
close all;

gamma_values = [0.49, 0.5]; % Gamma values

% Create figure
figure;
hold on;

for gamma = gamma_values
    x = [0.75, 20];      %x(1) = lockdown, x(2) = duration

    options = optimoptions('gamultiobj','PopulationSize',60,...
              'ParetoFraction',0.7);
    lb = [0 0];
    ub = [1 100];

    [solution, ObjectiveValue] = gamultiobj(@(x) eindfunctie(x,
       gamma), 2,...
                                [],[],[],[],lb,ub,options);

    plot(ObjectiveValue(:,1), ObjectiveValue(:,2), 'o');
end

% Add xlabel, ylabel, and legend
```

```
xlabel('I_{max}');
ylabel('u_{tot}');
legend(string(gamma_values), 'Location', 'northeast');

hold off;

% Create function
function laagste = mijnfunctie1(x, gamma)

    % Set Parameters
    Rt = 2;           % Transmission rate
    P = 10000000;     % Population
    I0 = 1000;        % Initial infected population
    tmax = 100;       % Length of simulation (in days)
    dt = 2;           % Time step (in days)

    % Initialize values
    t=0:dt:tmax;       % Time values (in days)
    nt = length(t);    % Number of timesteps
    S = zeros(1, nt); % Susceptible values
    I = zeros(1, nt); % Infected values
    R = zeros(1, nt); % Recovered values
    S(1) = P - I0;     % Initial value of S
    I(1) = I0;         % Initial value of I
    R(1) = 0;          % Initial value of R

    % Calculations
    beta = 0.6;        % Transmission rate (per day)

    t1 = 0:dt:tmax;
    t2 = t1 + x(2);

    aantal = 1;
    resultaten = zeros(aantal,3);

    Z = zeros(length(t1),length(t1));

    l=1;
    for j = 1:nt
        if (t1(j) < t2(j))
            u(t >= 0 & t < t1(j)) = 0;
            u(t >= t1(j) & t < t2(j)) = x(1);
```

```
        u(t >= t2(j)) = 0;

        resultaten(l,1) = t1(j);
        resultaten(l,2) = t2(j);

        for i = 2:nt
            dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;
                                % Change in S
            S(i) = S(i-1) + dS;
                                                    % Current S
                value
            if S(i) < 0; S(i) = 0; end

            dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i-1))
               *dt;   % Change in I
            I(i) = I(i-1)+dI;
                                                      % Current
                S value
            if I(i) < 0; I(i) = 0; end

            dR = (gamma*I(i-1))*dt;
                                                  % Change in R
            R(i) = R(i-1) + dR;
                                                    % Current R
                value

            Z(j,j) = I(i);
        end
        grootste = max(I(:));
        resultaten(l,3) = grootste;
        oppervlakte = trapz(t, I(:));
        resultaten(l,4) = oppervlakte;
        l=l+1;
    end
end

% Find minimum value of I_max
laagste = min(resultaten(:,4));
[X,Y] = find(Z == laagste);

% Find row in which value of column 4 is equal to laagste
rij = find(resultaten(:,4) == laagste, 1);
```

```
    % Starting day t1 that belongs to this row
    startdag = resultaten(rij,1);
end

% Create another function
function w = eindfunctie(x, gamma)
    laagste = mijnfunctie1(x, gamma);
    w(1) = laagste;
    w(2) = x(1)*x(2); %x(1) = lockdown, x(2) = duration
end
```

9 Appendix AC

```
clear;
close all;

gamma_values = [0.1, 0.14, 0.18]; % Values for gamma

% Create figure
figure;
hold on;

for gamma = gamma_values
    x = [0.75, 20];      %x(1) = lockdown, x(2) = duration

    options = optimoptions('gamultiobj','PopulationSize',60,...
              'ParetoFraction',0.7);
    lb = [0 0];
    ub = [1 150];

    [solution, ObjectiveValue] = gamultiobj(@(x) eindfunctie(x,
       gamma), 2,...
                                [],[],[],[],lb,ub,options);

    plot(ObjectiveValue(:,1), ObjectiveValue(:,2), 'o');
end

% Add xlabel, ylabel, and legend
xlabel('I_{max}');
ylabel('u_{tot}');
legend(string(gamma_values), 'Location', 'northeast');
```

```
hold off;

% Create function
function laagste = mijnfunctie1(x, gamma)

    % Set Parameters
    Rt = 2;           % Transmission rate
    P = 10000000;     % Population
    I0 = 1000;        % Initial infected population
    tmax = 100;       % Length of simulation (in days)
    dt = 2;           % Time step (in days)

    % Initialize values
    t=0:dt:tmax;       % Time values (in days)
    nt = length(t);    % Number of timesteps
    S = zeros(1, nt); % Susceptible values
    I = zeros(1, nt); % Infected values
    R = zeros(1, nt); % Recovered values
    S(1) = P - I0;     % Initial value of S
    I(1) = I0;         % Initial value of I
    R(1) = 0;          % Initial value of R

    % Calculations
    beta=0.6;          % Transmission rate (per day)

    t1 = 0:dt:tmax;
    t2 = t1 + x(2);

    aantal = 1;
    resultaten = zeros(aantal,3);

    Z = zeros(length(t1),length(t1));

    l=1;
    for j = 1:nt
        if (t1(j) < t2(j))
            u(t >= 0 & t < t1(j)) = 0;
            u(t >= t1(j) & t < t2(j)) = x(1);
            u(t >= t2(j)) = 0;

            resultaten(l,1) = t1(j);
            resultaten(l,2) = t2(j);
```

```
            for i = 2:nt
                dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;
                                % Change in S
                S(i) = S(i-1) + dS;
                                                    % Current S
                   value
                if S(i) < 0; S(i) = 0; end

                dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i-1))
                   *dt; % Change in I
                I(i) = I(i-1)+dI;
                                                      % Current E
                    value
                if I(i) < 0; I(i) = 0; end

                dR = (gamma*I(i-1))*dt;
                                                  % Change in R
                R(i) = R(i-1) + dR;
                                                    % Current R
                   value

                Z(j,j) = I(i);
            end
            grootste = max(I(:));
            resultaten(l,3) = grootste;
            oppervlakte = trapz(t, I(:));
            resultaten(l,4) = oppervlakte;
            l=l+1;
        end
    end

    % Find minimum value of I_max
    laagste = min(resultaten(:,4));
    [X,Y] = find(Z == laagste);

    % Find row in which value of column 4 is equal to laagste
    rij = find(resultaten(:,4) == laagste, 1);

    % Starting day t1 that belongs to this row
    startdag = resultaten(rij,1);
end
```

```
% Create another function
function w = eindfunctie(x, gamma)
    laagste = mijnfunctie1(x, gamma);
    w(1) = laagste;
    w(2) = x(1)*x(2); %x(1) = lockdown, x(2) = duration
end
```

10 Appendix AD

```
clear;
close all;

x = [0.6118, 57];      %x(1) = lockdown, x(2) = duration
w = eindfunctie(x);    %w(1) = minimum, w(2) = surface

options = optimoptions('gamultiobj','PopulationSize',60,...
          'ParetoFraction',0.7,'PlotFcn',@gaplotpareto);
lb = [0 0];
ub = [1 91];

[solution,ObjectiveValue] = gamultiobj(@eindfunctie,2,...
                            [],[],[],[],lb,ub,options);
% Create figure with extra plot of current I_max value
figure;
plot(ObjectiveValue(:,1), ObjectiveValue(:,2), 'o');
hold on;
x = 609020;     %I_max
y = 57*0.6118; % u_tot
plot(x,y, 'ro', 'MarkerSize', 5, 'LineWidth', 2)
hold off;
xlabel('I_{max}')
ylabel('u_{tot}')
legend('Pareto efficient solutions', 'Solution SIR-model')

function laagste = mijnfunctie1(x)

        % Set Parameters
        Rt = 2;          % Transmission rate
        P = 3400000;     % Population
        I0 = 1200;       % Initial infected population
        tmax = 91;       % Length of simulation (in days)
        dt = 2;          % Time step (in days)
```

```
% Initialize values
t=0:dt:tmax;       % Time values (in days)
nt = length(t);    % Number of timesteps
S = zeros(1, nt); % Susceptible values
I = zeros(1, nt); % Infected values
R = zeros(1, nt); % Recovered values
S(1) = P - I0;     % Initial value of S
I(1) = I0;         % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;     % Transmission rate (per day)
gamma = 1/7;       % Recovery rate (per day)

t1 = 0:dt:tmax;
t2 = t1 + x(2);

aantal = 1;
resultaten = zeros(aantal,3);

Z = zeros(length(t1),length(t1));

l=1;
for j = 1:nt
    if (t1(j) < t2(j))
        u(t >= 0 & t < t1(j)) = 0;
        u(t >= t1(j) & t < t2(j)) = x(1);
        u(t >= t2(j)) = 0;

        resultaten(l,1) = t1(j);
        resultaten(l,2) = t2(j);

        for i = 2:nt
            dS = (-beta*(1-u(i))*I(i-1)*S(i-1)/P)*dt;
                              % Change in S
            S(i) = S(i-1) + dS;
                                                    %
               Current S value
            if S(i) < 0; S(i) = 0; end

            dI = (beta*(1-u(i))*I(i-1)*S(i-1)/P-gamma*I(i
```

```
                    -1))*dt;  % Change in I
                    I(i) = I(i-1)+dI;
                                                        %
                    Current I value
                    if I(i) < 0; I(i) = 0; end

                    dR = (gamma*I(i-1))*dt;
                                                    % Change in
                    R
                    R(i) = R(i-1) + dR;
                                                        %
                    Current R value

                    Z(j,j) = I(i);
                end
                grootste = max(I(:));
                resultaten(l,3) = grootste;
                oppervlakte = trapz(t, I(:));
                resultaten(l,4) = oppervlakte;
                l=l+1;
            end
        end

        % Find minimum value of I_max
        laagste = min(resultaten(:,3));
        [X,Y] = find(Z == laagste);

        % Find row in which value of column 4 is equal to laagste
        rij = find(resultaten(:,3) == laagste, 1);

        % Starting day t1 that belongs to this row
        startdag = resultaten(rij,1);
end

% Create another function
function w = eindfunctie(x)
    laagste = mijnfunctie1(x);
    w(1) = laagste;
    w(2) = x(1)*x(2); %x(1) = lockdown, x(2) = duration
end
```

11 Appendix AE

```
clear;
close all;

x = [0.6118, 57];      %x(1) = lockdown, x(2) = duration
w = eindfunctie(x);    %w(1) = minimum, w(2) = surface

options = optimoptions('gamultiobj','PopulationSize',60,...
          'ParetoFraction',0.7,'PlotFcn',@gaplotpareto);
lb = [0 0];
ub = [1 250];

[solution,ObjectiveValue] = gamultiobj(@eindfunctie,2,...
                            [],[],[],[],lb,ub,options);
% Create figure
figure;
plot(ObjectiveValue(:,1), ObjectiveValue(:,2), 'o');
hold on;
x = 552630; %2487600; %I_max
y = 57*0.6118; % u_tot
plot(x,y, 'x', 'LineWidth', 2)
hold off;
xlabel('I_{max}')
ylabel('u_{tot}')
legend('Pareto efficient solutions', 'Solution SEIRS-model')

% Create function
function laagste = mijnfunctie1(x)

        % Set Parameters
        Rt = 2;          % Transmission rate
        P = 3400000;     % Population
        tmax = 250;      % Length of simulation (in days)
        dt = 0.5;        % Time step (in days)
        I0 = 1200;

        % Initialize values
        t=0:dt:tmax;       % Time values (in days)
        nt = length(t);    % Number of timesteps
        S = zeros(1, nt); % Susceptible values
        E = zeros(1, nt); % Incuation values
        I = zeros(1, nt); % Infected values
        R = zeros(1, nt); % Recovered values
```

```
S(1) = P - I0;     % Initial value of S
E(1) = 1200;       % Initial value of E
I(1) = 0;          % Initial value of I
R(1) = 0;          % Initial value of R

% Calculations
beta = 0.4137;       % Transmission rate (per day)
gamma = 1/7;         % Recovery rate (per day)
mu = 1/(76*365);     % Death and birthrate (per day)
omega = 1/365;       % Rate of loss of immunity (per day)
sigma = 1/5;         % Latency period (in days)
alpha = 1/(83*365); % Infection-induced death ratio (per
   day)

t1 = 0:dt:tmax;
t2 = t1 + x(2);

aantal = 1;
resultaten = zeros(aantal,3);

Z = zeros(length(t1),length(t1));

l=1;
for j = 1:nt
    if (t1(j) < t2(j))
        u(t >= 0 & t < t1(j)) = 0;
        u(t >= t1(j) & t < t2(j)) = x(1);
        u(t >= t2(j)) = 0;

        resultaten(l,1) = t1(j);
        resultaten(l,2) = t2(j);

        for i = 2:nt
            dS = (mu*P - mu*S(i-1) - (beta*(1-u(i))*I(i
               -1)*S(i-1)/P) + omega*R(i-1))*dt;    %
               Change in S
            S(i) = S(i-1) + dS;

               % Current S value
            if S(i) < 0; S(i) = 0; end

            dE = (beta*(1-u(i))*I(i-1)*S(i-1)/P - sigma*E
```

```
                    (i-1) - mu*E(i-1))*dt;              % Change
                     in E
                E(i) = E(i-1) + dE;

                    % Current E value
                if E(i) < 0; E(i) = 0; end

                dI = (sigma*E(i-1) - gamma*I(i-1) - (mu +
                    alpha)*I(i-1))*dt;                  %
                    Change in I
                I(i) = I(i-1) + dI;

                    % Current S value
                if I(i) < 0; I(i) = 0; end

                dR = (gamma*I(i-1) - mu*R(i-1) - omega*R(i-1)
                    )*dt;                               % Change
                     in R
                R(i) = R(i-1) + dR;

                    % Current R value
            end
            grootste = max(I(:));
            resultaten(l,3) = grootste;
            oppervlakte = trapz(t, I(:));
            resultaten(l,4) = oppervlakte;
            l=l+1;
        end
    end

    % Find minimum value of I_max
    laagste = min(resultaten(:,3));
    [X,Y] = find(Z == laagste);

    % Find row in which value of column 4 is equal to laagste
    rij = find(resultaten(:,3) == laagste, 1);

    % Starting day t1 that belongs to this row
    startdag = resultaten(rij,1);
end

% Create function
```

```
function w = eindfunctie(x)
    laagste = mijnfunctie1(x);
    w(1) = laagste;
    w(2) = x(1)*x(2); %x(1) = lockdown, x(2) = duration
end
```