Modeling the COVID-19 Coronavirus

Braam, Pieter A.

Award date:
2021

Eindhoven University of Technology

Modeling the COVID-19 Coronavirus

Bachelor Final Project

Pieter Braam

2WH40

July 19, 2021

Supervisor: Jan H.M. ten Thije Boonkkamp
Second corrector: Michiel E. Hochstenbach

Abstract

This study provides mathematical models to analyze the spreading of a pandemic to advance our knowledge of how to control the spread of viruses, such as the coronavirus. More specifically, this study analyzes the SIR and SEIRP models and the stability restrictions of their equilibria. In addition, the next generation matrix will be used to determine the reproduction number of both models. Furthermore, to better understand the spread of a disease in both models, multiple analytical methods will be used to present and compare plots. This study extends the models by introducing lockdowns and restriction free days for a certain period. Moreover, in the SIR model the maximum fraction of individuals having the disease and the maximum fraction of individuals having had the disease is determined, whereas the SEIRP model is extended by introducing the effects of vaccinations.

Contents

1 Introduction

In the beginning of 2020 the coronavirus started spreading around the world. Already in March of 2020 the World Health Organization (WHO) considered it to be a global pandemic [1]. Governments introduced lockdowns, social distancing rules and even curfews to fight the virus [2]. However, new developments such as new genetic changes of the coronavirus and even faster spreading variants stress difficulties to control the spread of the virus on national and global scales and therewith emphasis the importance of a profound understanding of the conditions under which its spreading can be controlled [3]. These conditions are based on predictions, which can be made by analyzing a disease describing mathematical model.

This study uses mathematical models to analyze the spreading of a pandemic to contribute to the understanding of how to control the spread of viruses, such as the coronavirus. Mathematical models describe a system and analyze the compartments within the system by using mathematical concepts [4]. Although mathematical models are simplifications of a system, they are widely used to study the behaviour of complex systems. Based on their behaviour predictions can be made by using real data. For example, in a mathematical model which analyzes the spreading of a virus the fraction of individuals having a disease and the effects of restrictions can be predicted.

Literature shows that many disease describing mathematical models have been developed to describe the spread of a pandemic. One of the simplest of these models is the susceptible-infected-recovered (SIR) model, where the total population is divided into three categories: the susceptibles, infected and recovered population. Other basic models divide the population into more categories, such as the SEIR and SEIRP model [5]. These basic models and extensions consist of a system of differential equations. Solutions of these systems are usually not known and hence analytical methods are used to analyze the behaviour of the models [6].

This study analyses the SIR and SEIRP models and discusses their usefulness in understanding and controlling the spread of the coronavirus. The former model is extended by introducing lockdowns and restriction free days. The latter model is extended by introducing the effects of vaccinations. In both models the (forward) Euler, 4-stage Runge-Kutta method and the Runge-Kutta Dormand-Prince method are used to provide analytical solutions and plots. The accuracy of these methods is discussed by comparing them with an exact solution of a special case of the SIR model with birth and death rates. The aim of this study is to asses which interventions are most effective to reduce the spreading of the coronavirus. Furthermore, the models help to predict the number of infected over time such that the occupation rate of hospitals can be determined. Moreover, the findings of this study may be useful for governments to decide which restrictions should be introduced and when this should happen to reduce the spreading of the coronavirus.

The remainder of this study is structured as follows: First, chapter 2 discusses the SIR model, including its reproduction number, the maximum fraction of infected and the stability of its equilibrium points. Then, numerical methods are used to give plots and the model is extended by introducing lockdowns, restriction free days and birth and death rates. Thereafter, the methods used are compared. Chapter 3 discusses the SEIRP model. Consistent with the SIR model, the stability of its equilibrium points are determined and numerical methods are used to illustrate the spreading of the coronavirus. This model is then extended by adding vaccination rates. Finally, in chapter 4 conclusions are drawn and discussed based on previous findings.

2 The SIR model

This chapter discusses the SIR model. First the description of the model will be given. Secondly, the parameters in this model are determined. The maximum fraction of infected and the fraction of the population that will be infected is determined next. Then the stability of the system is discussed. After that, numerical methods are used to give plots of the system. Thereafter, the SIR model is extended by introducing lockdowns and by considering the SIR model with birth and death rates. Lastly, the accuracy of the methods used to provide plots are assessed and compared with an exact solution.

2.1 Three categories

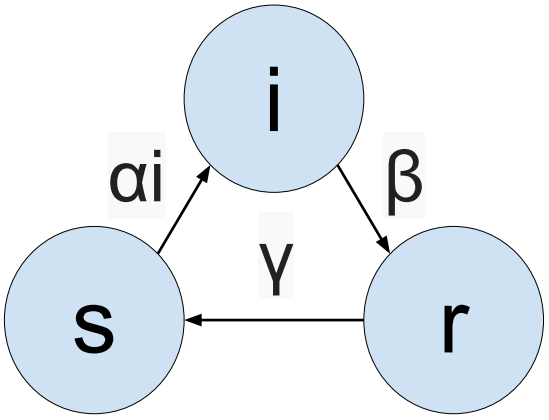


Figure 1: A flowchart of the SIR model.

One of the easiest ways to model the spread of the coronavirus is by means of the susceptible–infected–recovered (SIR) model. In this model the total population at time instant $t \geq 0$ days is divided into three categories: the population that can be infected with the virus called susceptibles, the infected population and the recovered population. Since each individual is assigned one of these labels, the categories are in epidemiology also known to be compartments. In the model it is assumed that the total population of N individuals stays the same, i.e., the SIR model is a closed system. Let us assume $s(t)$ to be the fraction of susceptibles of the total population, $i(t)$ to be the fraction of infected and $r(t)$ the fraction that is recovered, with

$$0 \leq s(t) \leq 1, \quad 0 \leq i(t) \leq 1, \quad 0 \leq r(t) \leq 1. \tag{1}$$

Since these three groups form the total population, their sum equals 1 for all $t \geq 0$, i.e.,

$$s(t) + i(t) + r(t) = 1. \tag{2}$$

In Figure 1 a simple flowchart of this model is presented. Here, the susceptibles are the part of the population that do not have the virus, but are able to be infected with a certain positive probability α called the infection rate. The infected are the ones that have the virus, can not be infected a second time and can recover from the virus with the recovery rate β. If this recovery rate β is set to 0, then the infective population can not recover and will hence stay in the infective category. Hence, it is assumed that the recovery rate β is strictly positive. The rate of increase of the infected is proportional to the product between susceptibles and infected, because susceptibles can only be infected when they come into contact with infected. Since the coronavirus's genetics can change and there might be immunity loss, it is assumed that these recovered individuals can become susceptibles again with probability γ called the immunity loss rate [5]. For each of the parameters it is assumed that

$$0 < \alpha \leq 1, \quad 0 < \beta \leq 1, \quad 0 \leq \gamma \leq 1. \tag{3}$$

In each of the previous cases, when an individual moves to another compartment, its previous compartment decreases in size, which can be seen by the arrows illustrated in Figure 1. Moreover, since at the moment the virus starts spreading each individual is assigned a label of one of the above categories, it is assumed that $s(0) + i(0) + r(0) = 1$. By denoting $\underline{y}(t) = (s(t), i(t), r(t))^T$ and $\underline{y}' = \underline{f}(t, \underline{y})$, the following set of equations can be found:

$$\underline{y}' = \begin{cases} \dfrac{\mathrm{d}s(t)}{\mathrm{d}t} = -\alpha s(t)i(t) + \gamma r(t), & \text{(4a)} \\ \dfrac{\mathrm{d}i(t)}{\mathrm{d}t} = \alpha s(t)i(t) - \beta i(t), & \text{(4b)} \\ \dfrac{\mathrm{d}r(t)}{\mathrm{d}t} = \beta i(t) - \gamma r(t). & \text{(4c)} \end{cases}$$

Adding these equations yields

$$\frac{\mathrm{d}s(t)}{\mathrm{d}t} + \frac{\mathrm{d}i(t)}{\mathrm{d}t} + \frac{\mathrm{d}r(t)}{\mathrm{d}t} = 0,$$

and using the initial condition $s(0) + i(0) + r(0) = 1$ gives equation (2). For later use, let

$$S(t) = Ns(t),\ I(t) = Ni(t) \text{ and } R(t) = Nr(t), \tag{5}$$

which in combination with equation (2) results in

$$S(t) + I(t) + R(t) = N. \tag{6}$$

2.2 Reproduction number and the fraction of infected

Using the assumptions given in (1) and (3) and by using equation (4b), observe that

$$\frac{\mathrm{d}i(t)}{\mathrm{d}t} = \alpha s(t)i(t) - \beta i(t) \leq (\alpha - \beta)i(t).$$

In the latter expression, $i(t)$ has a positive value for all $t \geq 0$ which indicates that the fraction of infected decreases when

$$\alpha - \beta < 0 \iff R_0 := \frac{\alpha}{\beta} < 1, \tag{7}$$

where R_0 is the reproduction number, which is equal to the infection rate divided by the recovery rate and hence also known to be the average number of people an infective individual will infect [5]. One might also be interested in what portion of the population ends up having the virus. Under the assumption that the immunity loss $\gamma = 0$, this portion can be calculated. Using this assumption, one finds with help of the first two equations of (4) that

$$\frac{\mathrm{d}i}{\mathrm{d}s} = \frac{\alpha si - \beta i}{-\alpha si} = -1 + \frac{\beta}{\alpha s} \implies \mathrm{d}i = \left(-1 + \frac{\beta}{\alpha s}\right)\mathrm{d}s. \tag{8}$$

Integrating both sides yields the equation

$$i(t) + s(t) - \frac{\beta \ln s(t)}{\alpha} = i(0) + s(0) - \frac{\beta \ln s(0)}{\alpha}. \tag{9}$$

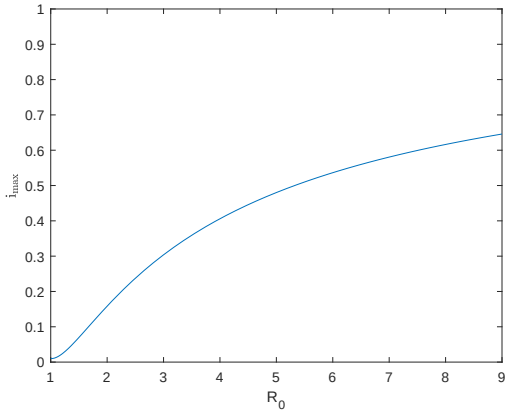


Figure 2: A plot of function $i_{\max}(R_0)$ given in equation (10) for $R_0 \geq 1$ and with initial values $s(0) = 0.99$, $i(0) = 0.01$ and $r(0) = 0$.

Equation (4b) tells us that $\frac{\mathrm{d}i}{\mathrm{d}t} = 0$ when $s = \frac{\beta}{\alpha}$. Using later numerical results described in section 2.4, one then finds that substituting this extremum in equation (9) yields the maximum fraction of infected in the SIR model

$$i_{\max}(R_0) = i(0) + s(0) - \frac{1}{R_0}\big(1 + \ln\big(R_0 \cdot s(0)\big)\big), \tag{10}$$

which holds for all $R_0 \geq 1$. When the virus started spreading almost the whole population was part of the susceptible compartment, only a small fraction was infected and no one was recovered. Hence, for numerical purposes it is for now assumed that 99 percent of the individuals in the total population are susceptibles, 1 percent is infected and no one is recovered, i.e., $s(0) = 0.99$, $i(0) = 0.01$ and $r(0) = 0$. A plot of the function $i_{\max}$ with these initial conditions can be seen in Figure 2. The maximum fraction of infected $i_{\max}$ appears to be increasing. In order to determine whether the function converges when $R_0 \to \infty$ we compute

$$\frac{\mathrm{d}i_{\max}}{\mathrm{d}R_0} = \frac{\ln(R_0 \cdot s(0))}{R_0^2} = 0 \implies R_0 = \frac{1}{s(0)},$$

which shows with help of Figure 2 that $i_{\max}$ has a minimum at

$$i_{\max}(R_0) = i(0) + s(0) - s(0)(1 + \ln(1)) = i(0),$$

and that $i_{\max}$ monotonically increases for $R_0 > 1$. As time progresses the fraction of infected $i(t)$ goes to 0, because there is no immunity loss and hence with help of equation (2) one finds $r(t_\infty) = 1 - s(t_\infty)$ where t_∞ is a large value of $t \geq 0$ such that $i(t_\infty) = 0$. Equation (9) for $t = t_\infty$ is given by

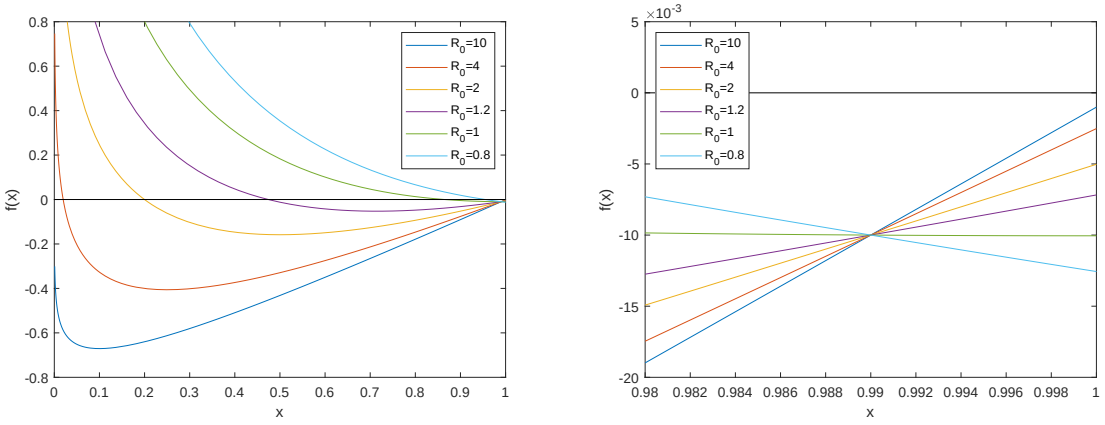
$$s(t_\infty) - \frac{\beta \ln s(t_\infty)}{\alpha} = i(0) + s(0) - \frac{\beta \ln s(0)}{\alpha}, \tag{11}$$

which can not be solved for $s(t_\infty)$ analytically. Similarly as above, it is for numerical purposes assumed that $s(0) = 0.99$, $i(0) = 0.01$ and $r(0) = 0$. Then equation (11) becomes

$$f(x) := x + \frac{1}{R_0} \ln\left(\frac{0.99}{x}\right) - 1 = 0, \tag{12}$$

where $x = s(t_\infty)$ and $f : (0, 1] \to \mathbb{R}$. In Figure 3 plots of function $f(x)$ can be seen for different values of R_0. In particular, notice that $f(s(0)) = f(0.99) = -0.01$ for all values of R_0, which can clearly be seen in Figure 3b. Since

$$\lim_{x \to 0} f(x) = +\infty, \; f(1) < 0, \; f'(x) = 1 - \frac{1}{R_0 x} = 0 \implies x = \frac{1}{R_0} > 0,$$



(a) $f(x)$ for x in range $(0, 1]$.

(b) $f(x)$ for x in range $[0.98, 1]$.

Figure 3: The function $f(x)$ given in equation (12) for different values of R_0.

and f is a continuous function for all values of $R_0 > 0$, function f has exactly one intersection point with the x-axis in its domain $(0, 1]$ and hence the Newton-Raphson method can be used to solve the nonlinear equation above. For different values of R_0 this method generates approximations of $s(t_\infty)$ using

$$s_n = s_{n-1} - \frac{f(s_{n-1})}{f'(s_{n-1})} = s_{n-1} - \frac{s_{n-1} + \frac{1}{R_0} \ln\left(\frac{0.99}{s_{n-1}}\right) - 1}{1 - \frac{1}{R_0 \cdot s_{n-1}}},$$

starting from a suitable initial value s_0. Observe in Figure 3a that if the starting value s_0 is chosen too large, the iteration might converge to a value on the x-axis outside the domain of the function f and hence s_0 should be small. Consequently, the fraction of the population that has once been infected when $t \to \infty$ can now be calculated, resulting in Figure 4. Hence, when $\alpha \gg \beta > 0$ the reproduction number R_0 is large and the total infected fraction is close to 1. Similarly, when R_0 is small the population gets infected slowly and gets recovered fast, resulting in the virus dying out fast. Therefore the fraction of the population that has been infected is small, which agrees with Figure 4. Lastly, when R_0 converges to 0, the virus dies out even faster and the numerical results tell us that the total fraction of the population that has been infected when $t \to 0$ then converges to 0.01, which can also be seen in Figure 4. Notice that this value was the initial fraction of infected of the population, as desired.

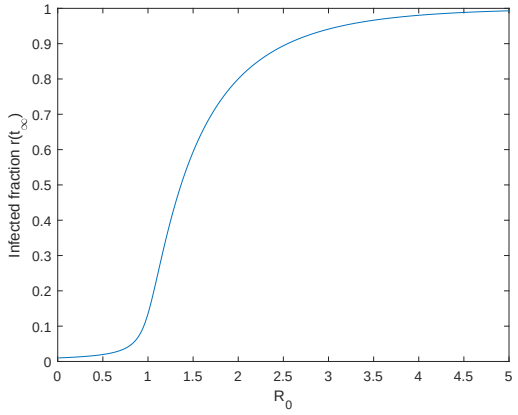


Figure 4: The total fraction of the population that has been infected when $t \to \infty$ for different values of R_0.

2.3 Stability

Under the assumption that $\gamma \neq 0$ and the assumptions given by (3), the equilibrium points of system (4) are by definition the points satisfying $\left(\frac{\mathrm{d}s(t)}{\mathrm{d}t}, \frac{\mathrm{d}i(t)}{\mathrm{d}t}, \frac{\mathrm{d}r(t)}{\mathrm{d}t}\right)^T = \underline{0} \in \mathbb{R}^3$. The second equation in the system then yields $i(t) = 0$ or $s(t) = \frac{\beta}{\alpha}$. The former is the trivial case where no one is infected, there is no spreading of the virus and the first equilibrium point is hence given by

$$\underline{y}_{\text{eq1}} = (1, 0, 0)^T. \tag{13}$$

Substituting the latter case in equation (2) and using equation (4c) with $\frac{\mathrm{d}r(t)}{\mathrm{d}t} = 0$, one finds the system

$$\begin{pmatrix} \beta & -\gamma \\ 1 & 1 \end{pmatrix} \begin{pmatrix} i(t) \\ r(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 - \frac{\beta}{\alpha} \end{pmatrix},$$

resulting in the second equilibrium point given by

$$\underline{y}_{\text{eq2}} = \left(\frac{\beta}{\alpha}, \frac{\gamma(\alpha - \beta)}{\alpha(\gamma + \beta)}, \frac{\beta(\alpha - \beta)}{\alpha(\gamma + \beta)}\right)^T. \tag{14}$$

Let us now calculate the Jacobian of $\underline{f}(\underline{y})$ given by (4). This Jacobian matrix is given by

$$J(\underline{y}) = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \frac{\partial f_1}{\partial y_3} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \frac{\partial f_2}{\partial y_3} \\ \frac{\partial f_3}{\partial y_1} & \frac{\partial f_3}{\partial y_2} & \frac{\partial f_3}{\partial y_3} \end{pmatrix} = \begin{pmatrix} -\alpha y_2 & -\alpha y_1 & \gamma \\ \alpha y_2 & \alpha y_1 - \beta & 0 \\ 0 & \beta & -\gamma \end{pmatrix}. \tag{15}$$

Since system (4) is closed, the sum of the entries in $J(y)$ is 0 and consequently $\lambda = 0$ is one of the eigenvalues. This derivation can be found in section 6.1.1. The two other eigenvalues are

$$\lambda = \frac{-\gamma - \alpha y_2 + \alpha y_1 - \beta \pm \sqrt{(\gamma + \alpha y_2 - \alpha y_1 + \beta)^2 + 4\alpha\gamma y_1 - 4(\alpha\gamma y_2 + \beta\gamma + \alpha\beta y_2)}}{2}. \tag{16}$$

A derivation for this can be found in section 6.1.2. Consequently, for the first equilibrium point $\underline{y}_{\text{eq1}}$ one finds the three eigenvalues $\lambda = 0$, $\lambda = -\gamma$ and $\lambda = \alpha - \beta$. This derivation can be found in section 6.1.3. An equilibrium solution $\underline{y}_{\text{eq}}$ is stable if for any neighborhood U of $\underline{y}_{\text{eq}}$ there exists a neighborhood V of this equilibrium point such that for any initial value $\underline{y}(0) \in V$ the corresponding solution $\underline{y}(t) \in U$ for all $t > 0$. Furthermore, an equilibrium point is stable when the real part of the eigenvalues of the Jacobian of the system are all non-positive. Otherwise an equilibrium point is unstable [6]. Hence, the first equilibrium point $\underline{y}_{\text{eq1}}$ is only stable when $\beta \geq \alpha$, i.e., when the recovery rate is larger than the infection rate. Similarly, the second equilibrium point $\underline{y}_{\text{eq2}}$ has the three eigenvalues $\lambda = 0$ and

$$\lambda = \frac{-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} \pm \sqrt{\frac{\gamma^2}{(\gamma+\beta)^2}\alpha^2 + \left(-\frac{2\beta\gamma^2}{(\gamma+\beta)^2} - \frac{2\gamma^2}{(\gamma+\beta)} - \frac{4\beta\gamma}{\gamma+\beta}\right)\alpha + \left(\gamma^2 + \frac{\gamma^2\beta^2}{(\gamma+\beta)^2} + \frac{2\beta\gamma^2}{\gamma+\beta} + \frac{4\beta^2\gamma}{\gamma+\beta}\right)}}{2}.$$

Moreover, this equilibrium point is stable when $\alpha \geq \beta$, i.e., when the infection rate is larger than the recovery rate. A derivation for these results can be found in section 6.1.4. Notice that the reproduction number R_0 as given in equation (7) can be related to these stability results. Equilibrium point $\underline{y}_{\text{eq1}}$ is stable when $R_0 \leq 1$ and equilibrium point $\underline{y}_{\text{eq2}}$ is stable when $R_0 \geq 1$.

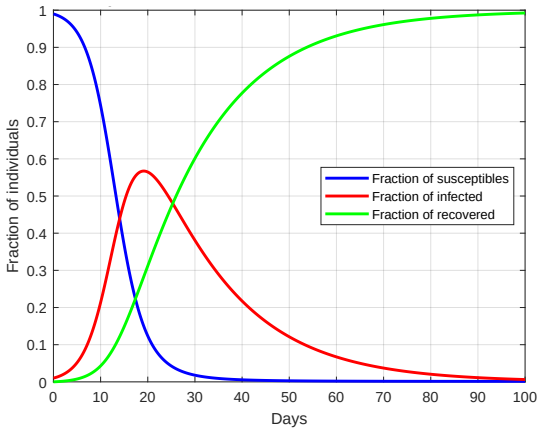


Figure 5: The first 100 days of the basic SIR model with parameters $\alpha = 0.4$, $\beta = 0.06$, $\gamma = 0.00$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$.

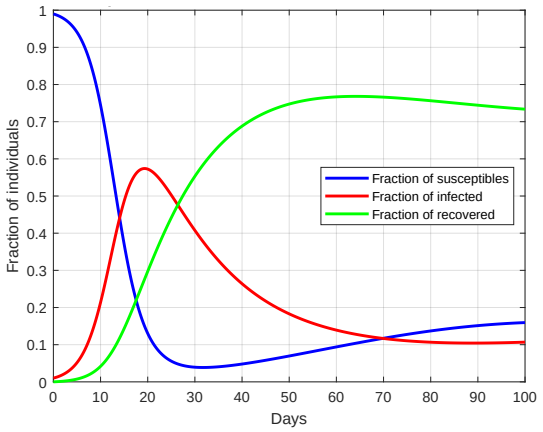


Figure 6: The first 100 days of the basic SIR model with parameters $\alpha = 0.4$, $\beta = 0.06$, $\gamma = 0.01$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$.

2.4 Numerical methods

One way of determining a graph of the solution of the system is by using the numerical (forward) Euler method. Let $h > 0$ denote the timestep and for $n = 0, 1, 2, ...$ let $t_n = nh$. Then the (forward) Euler method is given by

$$\underline{y}_{n+1} = \underline{y}_n + h\underline{f}(t_n, \underline{y}_n),$$

where $\underline{y}_n \approx \underline{y}(t_n)$, which is the most basic explicit method for systems of ordinary differential equations. Applying this method to the equations given by (4) yields

$$s_{n+1} = s_n + h\left(-\alpha s_n i_n + \gamma r_n\right),$$

$$i_{n+1} = i_n + h\left(\alpha s_n i_n - \beta i_n\right),$$

$$r_{n+1} = r_n + h\left(\beta i_n - \gamma r_n\right),$$

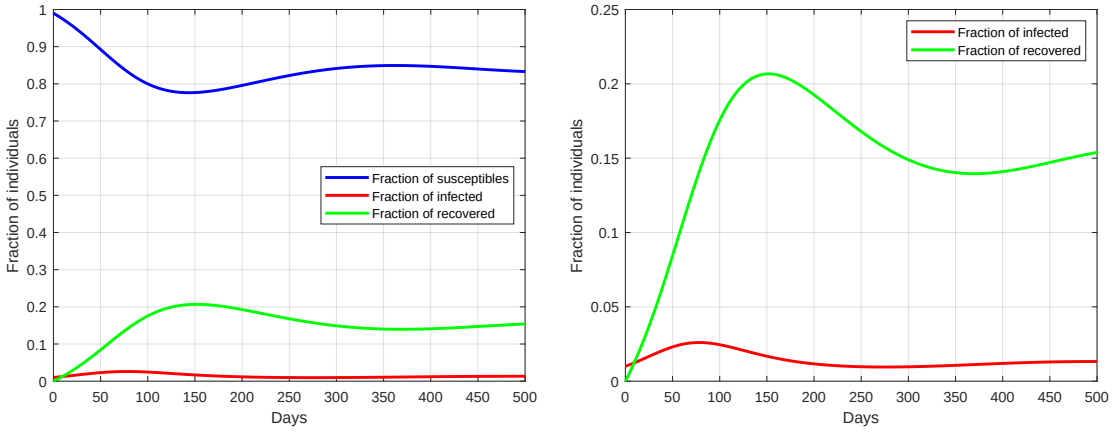
where $s_n \approx s(t_n)$, $i_n \approx i(t_n)$ and $r_n \approx r(t_n)$. Using this method and the software Matlab, plots of the SIR model can be found given in Figures 5, 6, 7a, 7b, 8a and 8b. It is assumed that at day 0 one percent of the population is infected and the rest are susceptibles, i.e., the initial conditions are given by $s(0) = 0.99$, $i(0) = 0.01$ and $r(0) = 0$. The first two figures illustrate the distribution of the three populations using the SIR model with parameters $\alpha = 0.40$ and $\beta = 0.06$ for the first 100 days, resulting in the reproduction number $R_0 = \frac{20}{3}$. Figure 5 shows the spreading of the virus without immunity loss after recovering, while in Figure 6 the immunity loss rate is assumed to be $\gamma = 0.01$. In the first days of Figure 5 the fraction of susceptibles decreases rapidly and converges towards zero, while the fraction of infected increases. Around the twentieth day the infected fraction of the population starts decreasing, while the recovered fraction monotonically increases. Computing the maximum fraction of infected, using parameters given in Figure 5, from equation (10) yields

$$i_{\max} = 0.01 + 0.99 - \frac{0.06}{0.40}\left(1 + \ln\left(\frac{0.40 \cdot 0.99}{0.06}\right)\right) \approx 0.5669,$$

which agrees with the maximum fraction of infected that can be seen in this figure.

The first days in the plot of the SIR model with immunity loss in Figure 6 behave similarly as the previous case, but after a while the fraction of susceptibles slowly increases and the recovered fraction decreases. Since $\alpha \geq \beta$ the equilibrium point $\underline{y}_{\text{eq1}}$ is in this case not stable, but $\underline{y}_{\text{eq2}}$ given by equation (14) is a stable equilibrium point and hence the fraction of susceptibles, infected and recovered converge to this equilibrium point. The numerical results tell us that the fraction of susceptibles, infected and recovered population at time $t = 1000$ are approximately equal to $0.1500, 0.1214$ and 0.7286, respectively, which match the exact results given by

$$\underline{y}_{\text{eq2}} = \left(\frac{0.06}{0.40}, \frac{0.01 \cdot (0.40 - 0.06)}{0.40 \cdot (0.01 + 0.06)}, \frac{0.06 \cdot (0.40 - 0.06)}{0.40 \cdot (0.01 + 0.06)} \right)^T \approx (0.1500, 0.1214, 0.7286)^T .$$



(a) The fraction of susceptibles, infected and recovered.

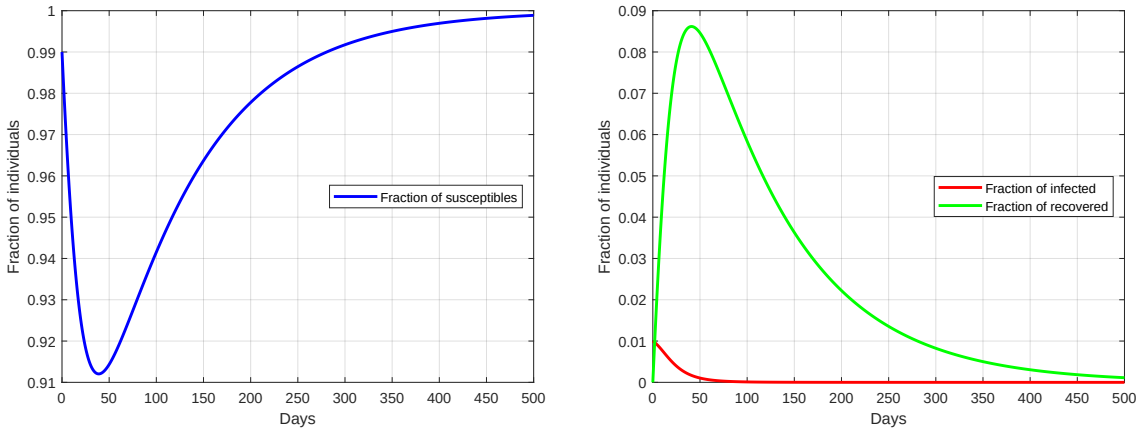
(b) The fraction of infected and recovered.

Figure 7: The first 500 days of the basic SIR model with parameters $\alpha = 0.15$, $\beta = 0.125$, $\gamma = 0.01$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$.

In reality the reproduction number R_0 is usually much closer to 1 and hence the value of α is closer to the value of β, resulting in a smaller fraction of infected than in the previous figures at any $t > 0$. This can clearly be seen in Figure 7a, where the first 500 days of the SIR model with parameters $\alpha = 0.15$, $\beta = 0.125$ and $\gamma = 0.01$ can be seen with the same timestep and initial conditions as in the previous figures. A zoom in on the fraction of infected and recovered of this figure can be seen in Figure 7b. In this case the reproduction number is $R_0 = \frac{6}{5}$, which is much closer to 1 and using equation (10) one finds that the maximum fraction of infected using these parameters is

$$i_{\max} = 0.01 + 0.99 - \frac{0.125}{0.15} \left(1 + \ln \left(\frac{0.15 \cdot 0.99}{0.125} \right) \right) \approx 0.0231,$$

which agrees with the value that can be seen in Figure 7b. In Figures 8a and 8b the first 500 days of SIR model with parameters $\alpha = 0.5$, $\beta = 0.51$ and $\gamma = 0.01$ can be seen. In this case the recovery rate β is higher than the infection rate α and hence equilibrium point $\underline{y}_{\text{eq1}}$ given by (13) is stable, i.e., the initial values converge to this point and consequently, the fraction of susceptibles converges to 1 and the virus dies out, which can clearly be seen in the figures.



(a) Fraction of susceptibles.

(b) Fraction of infected and recovered.

Figure 8: The first 500 days of the basic SIR model with parameters $\alpha = 0.5$, $\beta = 0.51$, $\gamma = 0.01$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$.

A more accurate numerical method is the so-called Runge-Kutta Dormand–Prince (RKDP) method. This method has a default method in Matlab (ode45) and can hence easily be used to compute solutions of system (4). Similarly as in the (forward) Euler method, let $h_n > 0$ denote the timestep and for $n = 0, 1, 2, ...$ let $t_n = nh_n$. Then this method gives a fourth order consistent approximation

$$\underline{y}_{n+1} = \underline{y}_n + h\left(\frac{35}{384}\underline{k}_{n,1} + \frac{500}{1113}\underline{k}_{n,3} + \frac{125}{192}\underline{k}_{n,4} - \frac{2187}{6784}\underline{k}_{n,5} + \frac{11}{84}\underline{k}_{n,6}\right),$$

and a fifth order consistent approximation given by

$$\tilde{\underline{y}}_{n+1} = \tilde{\underline{y}}_n + h\left(\frac{5179}{57600}\underline{k}_{n,1} + \frac{7571}{16695}\underline{k}_{n,3} + \frac{393}{640}\underline{k}_{n,4} - \frac{92097}{339200}\underline{k}_{n,5} + \frac{187}{2100}\underline{k}_{n,6} + \frac{1}{40}\underline{k}_{n,7}\right),$$

with

$$\underline{k}_{n,1} = \underline{f}(t_n, \underline{y}_n),$$

$$\underline{k}_{n,2} = \underline{f}\left(t_n + \frac{h}{5}, \underline{y}_n + \frac{h}{5}\underline{k}_{n,1}\right),$$

$$\underline{k}_{n,3} = \underline{f}\left(t_n + \frac{3h}{10}, \underline{y}_n + \frac{3h}{40}\underline{k}_{n,1} + \frac{9h}{40}\underline{k}_{n,2}\right),$$

$$\underline{k}_{n,4} = \underline{f}\left(t_n + \frac{4h}{5}, \underline{y}_n + \frac{44h}{45}\underline{k}_{n,1} - \frac{56h}{15}\underline{k}_{n,2} + \frac{32h}{9}\underline{k}_{n,3}\right),$$

$$\underline{k}_{n,5} = \underline{f}\left(t_n + \frac{8h}{9}, \underline{y}_n + \frac{19372h}{6561}\underline{k}_{n,1} - \frac{25360h}{2187}\underline{k}_{n,2} + \frac{64448h}{6561}\underline{k}_{n,3} - \frac{212h}{729}\underline{k}_{n,4}\right),$$

$$\underline{k}_{n,6} = \underline{f}\left(t_n + h, \underline{y}_n + \frac{9017h}{3168}\underline{k}_{n,1} - \frac{355h}{33}\underline{k}_{n,2} - \frac{46732h}{5247}\underline{k}_{n,3} + \frac{49h}{176}\underline{k}_{n,4} - \frac{5103h}{18656}\underline{k}_{n,5}\right),$$

$$\underline{k}_{n,7} = \underline{f}\left(t_n + h, \underline{y}_n + \frac{35h}{384}\underline{k}_{n,1} + \frac{500h}{1113}\underline{k}_{n,3} + \frac{125h}{192}\underline{k}_{n,4} - \frac{2187h}{6784}\underline{k}_{n,5} + \frac{11h}{84}\underline{k}_{n,6}\right).$$

0	0	0	0	0	0	0	0
$\frac{1}{5}$	$\frac{1}{5}$	0	0	0	0	0	0
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$	0	0	0	0	0
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$	0	0	0	0
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$	0	0	0
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$\frac{-5103}{18656}$	0	0
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Table 1: Butcher array for the Runge-Kutta Dormand-Prince method.

Notice that $\underline{k}_{n,2}$ is not implemented in both solutions, but is used to calculate $\underline{k}_{n,i}$ for $i \in \{3,4,5,6,7\}$. This Runge-Kutta method is explicit, which can be seen in the Butcher array in Table 1. Here, the last but one row gives the fourth order accurate solution and the last row gives the fifth order accurate solution. The difference between these two solutions is given by

$$\left|\underline{\tilde{y}}_{n+1} - \underline{y}_{n+1}\right| = \left|\frac{71}{57600}\underline{k}_{n,1} - \frac{71}{16695}\underline{k}_{n,3} + \frac{71}{1920}\underline{k}_{n,4} - \frac{17253}{339200}\underline{k}_{n,5} + \frac{22}{525}\underline{k}_{n,6} - \frac{1}{40}\underline{k}_{n,7}\right|,$$

which is the error estimate for the fourth order solution. Moreover, this method makes use of stepsize control, i.e., the value of h_{n+1} for each $n = 1, 2, ...$ in this method is determined by calculating

$$h_{n+1} = h_n \sqrt[5]{\frac{\epsilon h_n}{2|\underline{\tilde{y}}_{n+1} - \underline{y}_{n+1}|}}, \tag{17}$$

where the ϵ is a small tolerance term [7]. Using Matlab, this method yields approximately the same figures with the respective parameters as given in Figures 5, 6, 7a, 7b, 8a and 8b. Furthermore, a phase plot of the SIR model with initial conditions $s(0) = 0.99, i(0) = 0.01$, $r(0) = 0.00$ and parameters $\alpha = 0.40$, $\beta = 0.06$, $\gamma = 0.01$ for the first 100 days can be calculated, resulting in Figure 9. Notice that the susceptible fraction of the population decreases and that the recovered fraction of the population increases in these first 100 days. Furthermore, the infected fraction of the population increases at first, but decreases after a while. This agrees with the results discussed before about Figure 6. Moreover, in Figure 10 the direction field of the susceptible fraction versus the infective fraction of the population of the basic SIR model with the same parameters and timestep can be seen. Additionally, the two equilibrium points (13) and (14) have been implemented and a couple of streamlines have been added. Notice that if the number of infected equals zero the number of susceptibles converges to 1. This is due to the fact that there are no infected and hence no one can become infected. Since $\gamma > 0$ the recovered fraction of the population slowly decreases and these individuals become susceptibles. Hence, the population categories converge to equilibrium point $\underline{y}_{\text{eq1}}$. Similarly, if the fraction of infected is not equal to zero, then the population categories converge to $\underline{y}_{\text{eq2}}$. Both results can in the figure be seen by the direction of the arrows going to the equilibrium points illustrated by the red points.

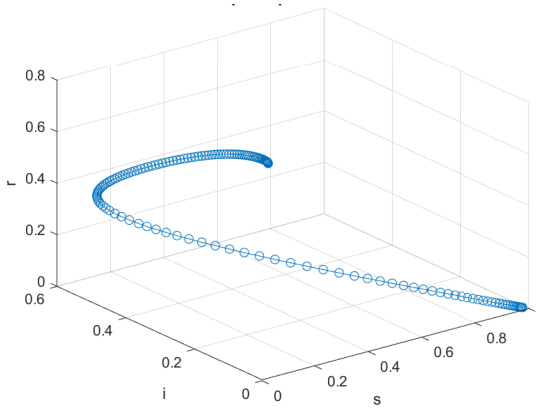


Figure 9: A phase plot of the susceptible, infective and recovered fraction of the total population of the basic SIR model with parameters $\alpha = 0.40$, $\beta = 0.06$ $\gamma = 0.01$ and timestep $h = 0.01$ for the first 100 days.

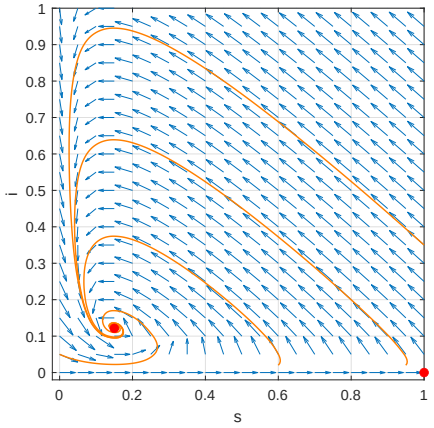


Figure 10: The direction field of the susceptible fraction versus the infective fraction of the population of the basic SIR model with parameters $\alpha = 0.40$, $\beta = 0.06$, $\gamma = 0.01$ and timestep $h = 0.01$ for the first 100 days. The equilibrium points given by equations (13) and (14) have been included in the plot just as streamlines starting in points (0,0.05), (0.6, 0.02), (0.95,0.02), (1,0.35), where the first number indicates the value of $s(t)$ and the second the value of $i(t)$.

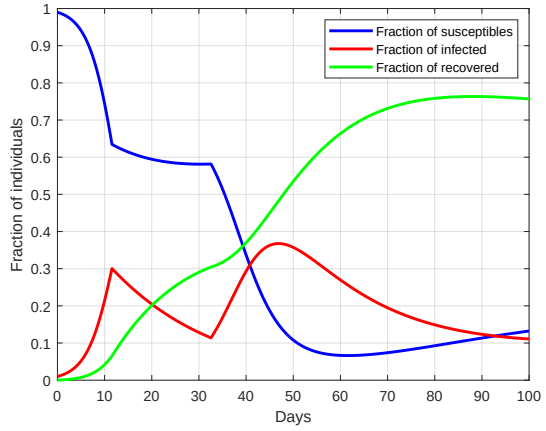


Figure 11: The basic SIR model with parameters $\alpha = 0.4$, $\beta = 0.06$, $\gamma = 0.01$, timestep $h = 0.01$, initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$ and a single lockdown period of 21 days that activates when 30% of the population is infected. The parameters during the lockdown are $\alpha = 0.04$, $\beta = 0.07$ and $\gamma = 0.01$.

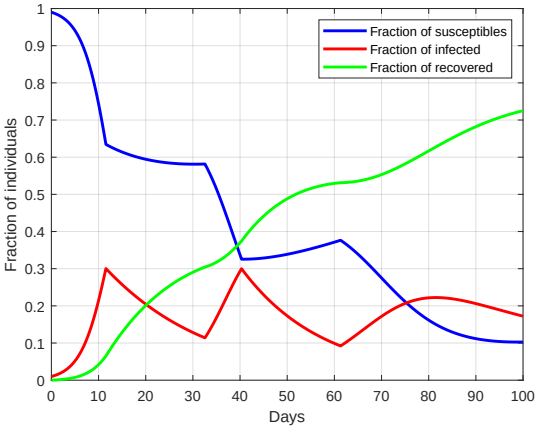


Figure 12: The basic SIR model with parameters $\alpha = 0.4$, $\beta = 0.06$, $\gamma = 0.01$, timestep $h = 0.01$, initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$ and a repeated lockdown period of 21 days that activates when 30% of the population is infected. The parameters during the lockdown are $\alpha = 0.04$, $\beta = 0.07$ and $\gamma = 0.01$.

2.5 The SIR model with a lockdown

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 2: Butcher array for the 4-stage Runge-Kutta method.

A third method that can be used to give plots of the solution of system (4) is the 4-stage Runge-Kutta method, which using Matlab results in approximately the same standard population distributions as given in Figures 5, 6, 7a, 7b, 8a and 8b. Similarly as in the (forward) Euler method, let $h > 0$ denote the timestep and for $n = 0, 1, 2, ...$ let $t_n = nh$. Then this method has the form

$$\underline{y}_{n+1} = \underline{y}_n + \frac{h}{6}\left(\underline{k}_{n,1} + 2\underline{k}_{n,2} + 2\underline{k}_{n,3} + \underline{k}_{n,4}\right),$$

with

$$\underline{k}_{n,1} = \underline{f}(t_n, \underline{y}_n),$$

$$\underline{k}_{n,2} = \underline{f}\left(t_n + \frac{h}{2}, \underline{y}_n + \frac{h}{2}\underline{k}_{n,1}\right),$$

$$\underline{k}_{n,3} = \underline{f}\left(t_n + \frac{h}{2}, \underline{y}_n + \frac{h}{2}\underline{k}_{n,2}\right),$$

$$\underline{k}_{n,4} = \underline{f}\left(t_n + h, \underline{y}_n + h\underline{k}_{n,3}\right),$$

where $\underline{f} = \left(\frac{\mathrm{d}s}{\mathrm{d}t}, \frac{\mathrm{d}i}{\mathrm{d}t}, \frac{\mathrm{d}r}{\mathrm{d}t}\right)^T$ as given by the equations in 4, $\underline{y}_n = (s_n, i_n, r_n)^T$ and $s_n \approx s(t_n)$, $i_n \approx i(t_n)$, $r_n \approx r(t_n)$. Its Butcher array is given in Table 2.

The Matlab model of this method can easily be modified in such a way that a lockdown can be implemented. This is done by changing the values of α, β and γ for a certain period of time and changing the parameters back to their original value after this period. For numerical purposes, assume that during the lockdown the infection rate α reduces to only 10 percent of its original value, because the spread of the virus is much slower during a lockdown due to less contact between individuals. Furthermore, assume that the recovery rate β increases slightly, since in hospitals there is more help provided, and assume that the lockdown takes three weeks. Lastly, the immunity loss is assumed to remain the same, since a lockdown does in general not influence the immune system. In Figure 11 an example of such an implementation of a lockdown can be seen with initial parameters $\alpha = 0.4$, $\beta = 0.06$ and $\gamma = 0.01$. The first few days of this population distribution are the same as in Figure 6 which shows the population distribution with the same parameters without a lockdown, but when 30% of the population is infected a three-week lockdown is entered where the parameters are changed to $\alpha = 0.04$, $\beta = 0.07$ and $\gamma = 0.01$. At this moment the rate at which the susceptibles become infected decreases. When the lockdown ends, which is exactly three weeks after it started, the virus develops in a similar manner as in Figure 6.

In reality, hospitals are not able to contain more than 1% of the population, but for insightful purposes assume that hospitals can maximally contain 30% of the population. Then a single lockdown as given in Figure 11 won't suffice. Hence a second lockdown needs to be implemented which can be done by slightly adjusting the model, resulting in Figure 12. Here, the number of infected stays below 30% of the population, as desired. As discussed in the previous section, in general the fraction of infected is much lower and lockdowns can last for much longer than three weeks. However, the same model with different parameters can be used resulting in similar results.

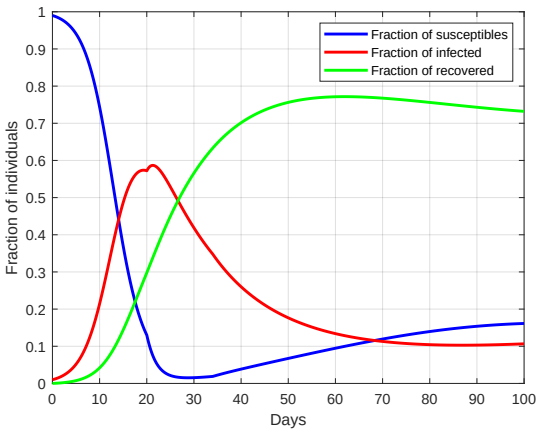


Figure 13: The basic SIR model with parameters $\alpha = 0.4$, $\beta = 0.06$, $\gamma = 0.01$, timestep $h = 0.01$, initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$ and a 14 days period starting at day 20 where the value of α is doubled.

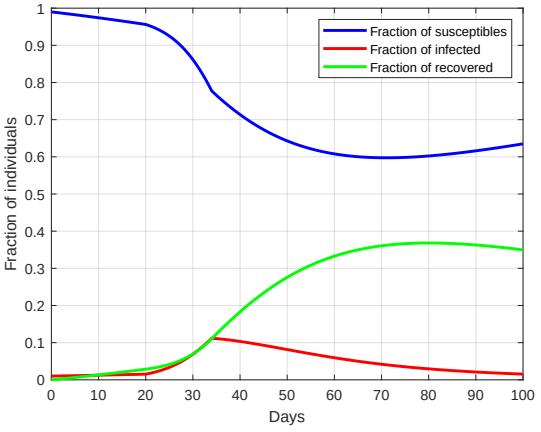


Figure 14: The basic SIR model with parameters $\alpha = 0.15$, $\beta = 0.125$, $\gamma = 0.01$, timestep $h = 0.01$, initial condition $\underline{y}(0) = (0.99, 0.01, 0)^T$ and a two-week period starting at day 20 where the value of α is doubled.

Instead of a lockdown, one can reverse the idea by canceling all rules regarding the virus for a certain period called '(restriction) free days'. Then people will come into contact more, which causes the infection rate α to increase. For numerical purposes, let $\alpha = 0.4$, $\beta = 0.06$ and $\gamma = 0.01$. Assume that this period starts on day 20, takes two weeks and that the infection rate α is doubled during this period. By making small changes to the previous Matlab model, one then finds Figure 13. Comparing this to Figure 6 without these free days, observe that such a period does not influence the distribution a lot and only provides a slight increase in the number of infected. However, in reality the reproduction number is close to 1, as discussed in section 2.4. Hence, assume that $\alpha = 0.15$, $\beta = 0.125$, $\gamma = 0.01$, that the free days period again starts on day 20, lasts two weeks and that the infection rate α is doubled. These parameters then give Figure 14. Comparing this with the first 100 days of Figures 7a and 7b, which have the same parameters, but without free days shows a large difference in the distribution of the compartments. In section 2.4 it was determined that the maximum fraction of infected using these parameters without a free days period is $i_{\max} \approx 0.0231$. When including these free days, the numerical model shows that the $i_{\max} \approx 0.1108$, which indicates the impact these free days have on the distribution of the virus.

2.6 The fatal SIR model

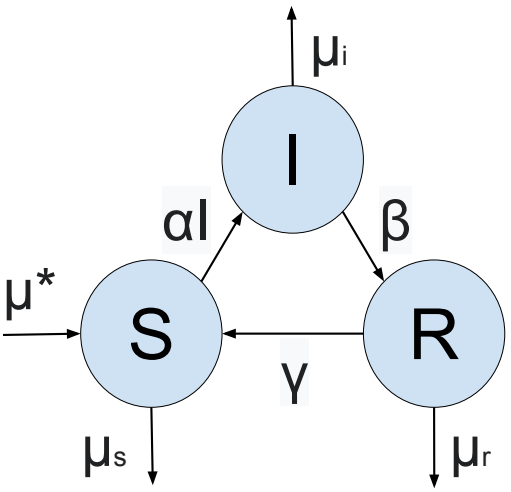


Figure 15: A flowchart of the fatal SIR model.

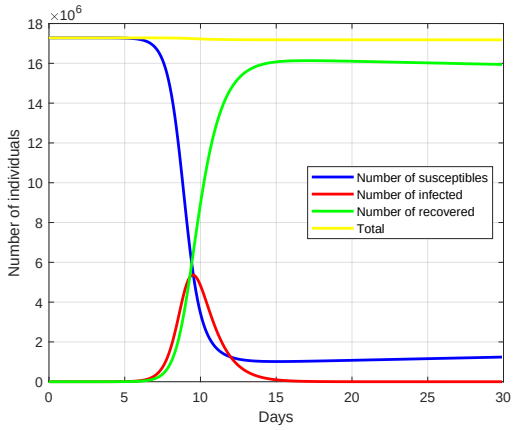
The SIR model can be slightly changed by adding rates of birth $\tilde{\mu}^\star$ per day and deathrates μ_s, μ_i and μ_r depending on the fraction of the susceptible, infective and recovered population, respectively, yielding the non-closed system

$$\underline{\tilde{y}}' = \begin{cases} \dfrac{\mathrm{d}s(t)}{\mathrm{d}t} = -\alpha s(t)i(t) + \gamma r(t) - \mu_s s(t) + \tilde{\mu}^\star, & \text{(18a)} \\ \dfrac{\mathrm{d}i(t)}{\mathrm{d}t} = \alpha s(t)i(t) - \beta i(t) - \mu_i i(t), & \text{(18b)} \\ \dfrac{\mathrm{d}r(t)}{\mathrm{d}t} = \beta i(t) - \gamma r(t) - \mu_r r(t), & \text{(18c)} \end{cases}$$

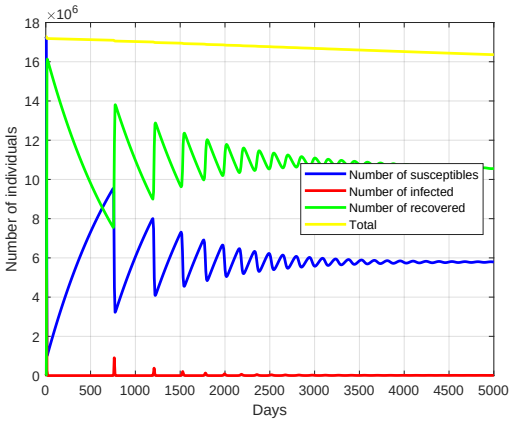
where $\underline{\tilde{y}}(t) = (s(t), i(t), r(t))^T$. Here it is assumed that newborns are susceptibles. Moreover, by using the definitions given in equation (5) one can transform system (18) to the following system:

$$\underline{\hat{y}}' = \begin{cases} \dfrac{\mathrm{d}S(t)}{\mathrm{d}t} = -\alpha S(t)I(t) + \gamma R(t) - \mu_s S(t) + \mu^\star, & \text{(19a)} \\ \dfrac{\mathrm{d}I(t)}{\mathrm{d}t} = \alpha S(t)I(t) - \beta I(t) - \mu_i I(t), & \text{(19b)} \\ \dfrac{\mathrm{d}R(t)}{\mathrm{d}t} = \beta I(t) - \gamma R(t) - \mu_r R(t), & \text{(19c)} \end{cases}$$

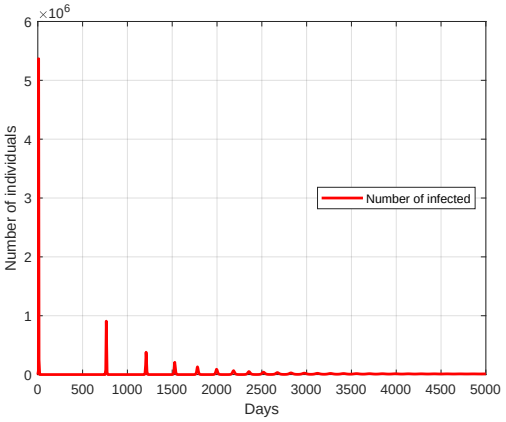
where $\underline{\hat{y}}(t) = (S(t), I(t), R(t))^T$, $\mu^\star = N\tilde{\mu}^\star$ and α is properly scaled. A flowchart of this system can be seen in Figure 15. The system can be used to model, e.g., the spreading of the coronavirus in the Netherlands. In 2020 the number of people in the Netherlands was about 17.28 million. For numerical purposes then assume that $\underline{\hat{y}}(0) = (1.728 \cdot 10^7, 1, 0)^T$. In 2018 approximately 169,000 babies were born, which is on average about 463 per day and hence consider $\mu^\star = 463$ [8]. According to [9] the number of Dutch citizens which have had the coronavirus was about 2.3 million on the first day of January 2021. Furthermore, according to [10] the number of Dutch coronavirus deaths up to that date was 13,422, resulting in an approximate mortality rate of $\frac{13{,}422}{2{,}300{,}000} \approx 0.005836$.



(a) A period of 30 days.



(b) A period of 5000 days.



(c) The number of infected over a period of 5000 days.

Figure 16: The fatal SIR model with parameters $\alpha = 1.728 \cdot 10^{-7}$, $\beta = 0.966$, $\gamma = 0.01$, $\mu^\star = 463$, $\mu_s = 0.00003389$, $\mu_i = 0.00586989$, $\mu_r = 0.00003389$, initial condition $\underline{y}(0) = (1.728 \cdot 10^7, 1, 0)$ and timestep $h = 0.1$.

Consequently, the approximated recovery rate is $\beta = 1 - \frac{13{,}422}{2{,}300{,}000} \approx 0.994164$. By assuming an average life expectancy of 81 years, which is approximately 29.5 thousand days [11] one finds

$$\mu_s = 0.00003389,\ \mu_i = 0.005836 + 0.00003389 = 0.00586989,\ \mu_r = 0.00003389,$$

because $\frac{1}{29{,}500} \approx 0.00003389$. According to [12] it is not yet known how long it takes for an individual to become vulnerable for getting the coronavirus a second time after recovering and hence for numerical purposes it is assumed that $\gamma = 0.001$. Lastly, by scaling α with respect to the total population and letting it equal to $1.728 \cdot 10^{-7}$, one can apply forward Euler in Matlab in a similar way as described in section 2.4 using timestep $h = 0.1$ over a period of 300 days, resulting in Figure 16a. Notice that the three compartments of the population in these first 300 days appear to behave similarly to the numerical results of the basic SIR model described in section 2.4. Furthermore, in this figure it appears that the total population does not decrease rapidly, due to the mortality rate of the virus being less than 1 percent. In Figures 16b and 16c plots of the fatal SIR model with the same parameters, but over a period of 5000 days, which is approximately 13.7 years, can be seen. The numerical results show that during these first 5000 days the total population has been decreased by almost 1 million. However, in reality a vaccine would in the meantime have been developed, resulting in a large decrease of the mortality rate.

2.7 Comparing methods

In previous sections three different methods have been used: the (forward) Euler method, the 4-stage Runge-Kutta method and the Runge-Kutta Dormand-Prince method. Notice that the (forward) Euler method can also be seen as a Runge-Kutta method. Runge-Kutta methods preserve linear invariance and hence

$$s(t_n) + i(t_n) + r(t_n) = 1.$$

This derivation can be found in Theorem 2 of section 6.1.1. In order determine the (approximated) errors each of these methods make, it is easier to compare their solutions with exact solutions. In particular, the fraction of infected will be analyzed. However, since exact solutions of the general SIR system (4) do not exist, system (18) with parameters $\gamma = 0$ and $\mu := \mu_s = \beta + \mu_i = \mu_r = \tilde{\mu}^\star$ will be used to compare the methods, yielding the system

$$\underline{\tilde{y}}' = \begin{cases} \dfrac{\mathrm{d}s(t)}{\mathrm{d}t} = -\alpha s(t)i(t) - \mu s(t) + \mu, & \text{(20a)} \\ \dfrac{\mathrm{d}i(t)}{\mathrm{d}t} = \alpha s(t)i(t) - \mu i(t), & \text{(20b)} \\ \dfrac{\mathrm{d}r(t)}{\mathrm{d}t} = \beta i(t) - \mu r(t). & \text{(20c)} \end{cases}$$

The system consisting of only the first two equations is the SIR model as described by Kermack and Mckendrick [13]. Exact solutions for the fraction of infected $i(t)$ for $t \geq 0$ in this system have been found in chapter 3 of [14], given by

$$i(t) = \frac{\lambda i(0)}{\alpha i(0) + (\lambda - \alpha i(0))\, e^{-\lambda t}} \text{ where } \lambda = \alpha - \mu + \alpha(s(0) + i(0) - 1).$$

For numerical purposes, let

$$\alpha = 0.3,\ \beta = 0.045,\ \mu = 0.2,\ s(0) = 0.99,\ i(0) = 0.01,\ r(0) = 0, \tag{21}$$

which at day $t = 200$ yield the fraction of infected

$$i(200) = \frac{0.001}{0.003 + 0.097 \cdot e^{-20}} \approx 0.333333311118679. \tag{22}$$

Table 3: Fraction of infected on day 200 of system (20) with parameters given in (21) using Forward Euler with the exact solution given by (22).

Timestep h	1	0.1	0.01
Fraction of infected	0.333333322270946	0.333333312524624	0.333333311262558
Error	$1.1152 \cdot 10^{-8}$	$1.4059 \cdot 10^{-9}$	$1.4387 \cdot 10^{-10}$

Table 4: Fraction of infected on day 200 of system (20) with parameters given in (21) using the 4-stage Runge-Kutta method with the exact solution given by (22).

Timestep h	1	0.1	0.01
Fraction of infected	0.333333311118362	0.333333311118679	0.333333311118679
Error	$3.1663 \cdot 10^{-13}$	$1.1102 \cdot 10^{-16}$	$4.4408 \cdot 10^{-16}$

This exact result can be compared to the fraction of infected on day 200 as calculated by the (forward) Euler method with the same parameters, resulting in Table 3. In this table it can be seen that a lower timestep yields a smaller error. More specifically, making the timestep ten times smaller yields an error which is approximately ten times smaller. This agrees with the global truncation error of the (forward) Euler method, which is $\mathcal{O}(h)$. Similarly, the fraction of infected on day 200 as calculated by the 4-stage Runge-Kutta method can be found, which results in Table 4. The global truncation error of this method is of order $\mathcal{O}(h^4)$. Notice that for each timestep this method yields a more accurate result than the (forward) Euler method. However, since Matlab only uses 16 digits of precision the error does not become smaller when reducing the timestep from $h = 0.1$ to $h = 0.01$. The error becomes even larger, but this is due to the rounding errors of Matlab.

Table 5: Fraction of infected on day 200 of system (20) with parameters given in (21) using the Runge-Kutta Dormand-Prince method with the exact solution given by (22).

Tolerance term ϵ	10^{-8}	10^{-12}	10^{-15}
Fraction of infected	0.333333310736543	0.333333311118595	0.333333311118679
Error	$3.8213 \cdot 10^{-10}$	$8.4210 \cdot 10^{-14}$	$2.7755 \cdot 10^{-16}$

The results of the fraction of infected at day 200 using the Runge-Kutta Dormand-Prince method can be seen in Table 5. As described in section 2.4, this method has a default method in Matlab called ODE45 and makes use of stepsize control, given in equation (17), where ϵ is the small tolerance term. When choosing a small tolerance term as in Table 5, this method yields smaller errors than in the Euler and 4-stage Runge-Kutta methods. However, similarly as in the 4-stage Runge-Kutta method, Matlab only uses 16 digits of precision and the error can hence not be made infinitely small by using Matlab.

3 The SEIRP model

This section discusses the SEIRP model. First a description of this model will be given. Its equilibrium point and the stability condition of this equilibrium point will be determined next. Thereafter, the reproduction number of the model will be determined by making use of a method using the next generation matrix. Then numerical methods are used to give plots of solutions of the system. Lastly, the model will be extended by introducing vaccinations.

3.1 Five categories

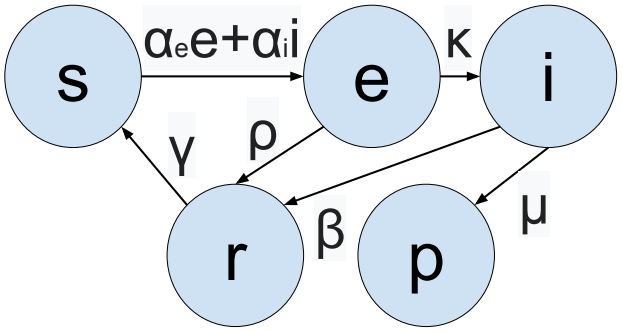


Figure 17: A flowchart of the SEIRP model.

Governments are usually interested in the mortality rate of a virus, which is not implemented in the basic SIR model. In section 2.6 this model was slightly elaborated by including birth and death processes, resulting in system (18). However, since this system is not closed, it is harder to analyze and provide plots. Secondly, when an individual is exposed to the virus, it might take days or even weeks for this individual to gain symptoms and to be able to infect others. Both of these issues are implemented in the so called SEIRP model, also called the fatal SEIR model [5]. In this model the population is divided into five categories. Similarly as in the SIR model, this model consists of the susceptible fraction $s(t)$, the infective fraction $i(t)$ and the recovered fraction $r(t)$. Furthermore, this model also includes the exposed fraction of the population $e(t)$, i.e., those who have been exposed to the virus, but do not have any symptoms and are hence not infected. The fifth category in this model is the fraction of the population $p(t)$ that have passed away due to the virus. Since these compartments form the total population, one finds

$$s(t) + e(t) + i(t) + r(t) + p(t) = 1. \tag{23}$$

The susceptible fraction of the population does not have the virus, but is exposed to the virus by the exposed population with probability α_e or by the infected population with probability α_i. These parameters are also called the contagion factors between the exposed population and the susceptibles and between the infected and susceptibles, respectively [5]. Note that in reality the first of these parameters is larger, since contact is usually not avoided with exposed individuals, because it is not known yet that these individuals have the virus. Furthermore, it is assumed that the exposed fraction of the population gets infected with probability κ, which can be seen as the parameter determining the speed at which exposed individuals get symptoms. They then recover from this state with probability ρ called the recovery rate of the exposed fraction of the population [5]. Since the virus has not fully penetrated for people in this compartment the chance of dying is assumed to be zero. However, the probability of passing away for infected is assumed to be μ. Furthermore, similarly as in the SIR model, their probability of recovering is β. Lastly, the recovered fraction $r(t)$ of the population become susceptibles again with the immunity loss rate and probability γ. In Figure 17 a flowchart of this model can be seen. For each of the parameters it is assumed that

$$0 \leq \alpha_e \leq 1, \quad 0 \leq \alpha_i \leq 1, \quad 0 < \beta \leq 1, \quad 0 < \gamma \leq 1, \quad 0 < \kappa \leq 1, \quad 0 < \rho \leq 1, \quad 0 < \mu \leq 1.$$

Furthermore, it is assumed that $\alpha_e + \alpha_i > 0$, since the disease would otherwise not be spreading. By denoting $\underline{y} = (s(t), e(t), i(t), r(t), p(t))^T$ and $\underline{y}' = \underline{f}(\underline{y})$, the previous assumptions result in the following set of equations:

$$\underline{y}' = \begin{cases} \dfrac{\mathrm{d}s(t)}{\mathrm{d}t} = -\alpha_e s(t)e(t) - \alpha_i s(t)i(t) + \gamma r(t), & \text{(24a)} \\ \dfrac{\mathrm{d}e(t)}{\mathrm{d}t} = \alpha_e s(t)e(t) + \alpha_i s(t)i(t) - \kappa e(t) - \rho e(t), & \text{(24b)} \\ \dfrac{\mathrm{d}i(t)}{\mathrm{d}t} = \kappa e(t) - \beta i(t) - \mu i(t), & \text{(24c)} \\ \dfrac{\mathrm{d}r(t)}{\mathrm{d}t} = \beta i(t) + \rho e(t) - \gamma r(t), & \text{(24d)} \\ \dfrac{\mathrm{d}p(t)}{\mathrm{d}t} = \mu i(t). & \text{(24e)} \end{cases}$$

Similarly as in the SIR model, one finds by adding these equations and using the initial condition $s(0) + e(0) + i(0) + r(0) + p(0) = 1$ equation (23).

3.2 Stability

Similarly as in the SIR model, assuming that the parameters are unequal to 0, the equilibrium points of system (24) are by definition the points satisfying $\left(\frac{\mathrm{d}s(t)}{\mathrm{d}t}, \frac{\mathrm{d}e(t)}{\mathrm{d}t}, \frac{\mathrm{d}i(t)}{\mathrm{d}t}, \frac{\mathrm{d}r(t)}{\mathrm{d}t}, \frac{\mathrm{d}p(t)}{\mathrm{d}t}\right)^T = \underline{0} \in \mathbb{R}^5$, yielding the only equilibrium point

$$\underline{y}_{\text{eq}} = (1 - p_\infty, 0, 0, 0, p_\infty)^T, \tag{25}$$

where $p_\infty \in [0, 1]$ is the passed away fraction of the population in transient state. Since in this equilibrium point the fraction of exposed and infected is 0, this is a disease-free equilibrium point (DFE). Let us now calculate the Jacobian of $\underline{f}(\underline{y})$ in system (24). This Jacobian matrix in the above equilibrium point is given by

$$J(\underline{y}_{\text{eq}}) = \begin{pmatrix} 0 & -\alpha_e(1-p_\infty) & -\alpha_i(1-p_\infty) & \gamma & 0 \\ 0 & \alpha_e(1-p_\infty) - \kappa - \rho & \alpha_i(1-p_\infty) & 0 & 0 \\ 0 & \kappa & -\beta - \mu & 0 & 0 \\ 0 & \rho & \beta & -\gamma & 0 \\ 0 & 0 & \mu & 0 & 0 \end{pmatrix}, \tag{26}$$

which has the two zero eigenvalues $\lambda_1 = \lambda_2 = 0$ and eigenvalues

$$\lambda_3 = -\gamma,\ \lambda_{4,5} = \frac{\alpha_e(1-p_\infty) - \kappa - \rho - \beta - \mu \pm \sqrt{(\beta + \mu + \alpha_e(1-p_\infty) - \kappa - \rho)^2 + 4\kappa\alpha_i}}{2}. \tag{27}$$

This derivation can be found in section 6.1.5. Clearly, the part inside the root of the last two eigenvalues is positive and hence each eigenvalue is real. Furthermore, the eigenvalue λ_4 with the positive root is the largest of these two eigenvalues. Therefore, assuming $p_\infty = 0$ the equilibrium point $\underline{y}_{\text{eq}}$ is stable when $\lambda_4 \leq 0$, i.e., when

$$\kappa\alpha_i \leq (\kappa + \rho - \alpha_e)(\beta + \mu). \tag{28}$$

This derivation can be found in section 6.1.6.

3.3 Reproduction number

Unlike in the SIR model, the reproduction number R_0 can not be found as easily as described at the start of section 2.2. In order to determine R_0 in the SEIRP model an alternative method needs to be used, described in [15]. First, the vector field $\underline{f}(\underline{y})$ given in (24) will be written in the form $\underline{f}(\underline{y}) = \underline{\mathcal{F}}(\underline{y}) + \underline{\mathcal{V}}(\underline{y})$. Here, $\underline{\mathcal{F}}$ consists of all new infection terms of $\underline{f}(\underline{y})$ and $\underline{\mathcal{V}}(\underline{y})$ of the remaining terms. Population shifts between infection compartments are not considered to be new infections. Hence one finds

$$\underline{\mathcal{F}} = \begin{pmatrix} 0 \\ \alpha_e s(t)e(t) + \alpha_i s(t)i(t) \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ and } \underline{\mathcal{V}} = \begin{pmatrix} -\alpha_e s(t)e(t) - \alpha_i s(t)i(t) + \gamma r(t) \\ -\kappa e(t) - \rho e(t) \\ \kappa e(t) - \beta i(t) - \mu i(t) \\ \beta i(t) + \rho e(t) - \gamma r(t) \\ \mu i(t) \end{pmatrix}.$$

The disease-free equilibrium point given in equation (25) satisfies each of the five conditions described in section 2 of [15]. These conditions can also be found in section 6.1.7. Consequently, one can make use of Lemma 1 described in the article, which states that the Jacobian of $\underline{\mathcal{F}}$ and $\underline{\mathcal{V}}$ in a decease free equilibrium $\underline{y}_{\text{eq}}$ can be written in the form

$$\nabla_{\underline{y}} \underline{\mathcal{F}}(\underline{y}_{\text{eq}}) = \begin{pmatrix} F & 0 \\ 0 & 0 \end{pmatrix} \text{ and } \nabla_{\underline{y}} \underline{\mathcal{V}}(\underline{y}_{\text{eq}}) = \begin{pmatrix} V & 0 \\ J_1 & J_2 \end{pmatrix},$$

where F and V are square matrices, F is nonnegative, V is nonsingular and both F and V only consist of the infection compartments. Using this lemma on the disease-free equilibrium point (25) yields the Jacobians

$$\nabla_{\underline{y}} \underline{\mathcal{F}}(\underline{y}_{\text{eq}}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_e(1-p_\infty) & \alpha_i(1-p_\infty) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\nabla_{\underline{y}} \underline{\mathcal{V}}(\underline{y}_{\text{eq}}) = \begin{pmatrix} 0 & -\alpha_e(1-p_\infty) & -\alpha_i(1-p_\infty) & \gamma & 0 \\ 0 & -\kappa-\rho & 0 & 0 & 0 \\ 0 & \kappa & -\beta-\mu & 0 & 0 \\ 0 & \rho & \beta & -\gamma & 0 \\ 0 & 0 & \mu & 0 & 0 \end{pmatrix}.$$

Notice that $\nabla_{\underline{y}} \underline{\mathcal{F}}(\underline{y}_{\text{eq}}) + \nabla_{\underline{y}} \underline{\mathcal{V}}(\underline{y}_{\text{eq}}) = J(\underline{y}_{\text{eq}})$ where $J(\underline{y})$ is the Jacobian given in equation (26), as desired. Matrices F and V only consist of the infection compartments, which are indicated by the boxes above. Consequently, one finds

$$F(\underline{y}_{\text{eq}}) = \begin{pmatrix} \alpha_e(1-p_\infty) & \alpha_i(1-p_\infty) \\ 0 & 0 \end{pmatrix} \text{ and } V(\underline{y}_{\text{eq}}) = \begin{pmatrix} -\kappa-\rho & 0 \\ \kappa & -\beta-\mu \end{pmatrix}.$$

Then one can make use of Theorem 2 in [15], which states that $R_0 = \rho(-FV^{-1})$, where ρ is the spectral radius of the so called 'next generation matrix' (NGM) $-FV^{-1}$, which is known to be the natural basis matrix of the reproduction number [16]. In our case, one finds the next generation matrix

$$-FV^{-1} = \frac{1}{(\kappa+\rho)(\beta+\mu)} \begin{pmatrix} \alpha_e(1-p_\infty)(\beta+\mu) + \kappa\alpha_i(1-p_\infty) & \alpha_i(1-p_\infty)(\kappa+\rho) \\ 0 & 0 \end{pmatrix},$$

which has spectral radius and reproduction number

$$R_0 = \rho(-FV^{-1}) = \frac{\alpha_e(1-p_\infty)(\beta+\mu) + \kappa\alpha_i(1-p_\infty)}{(\kappa+\rho)(\beta+\mu)}.$$

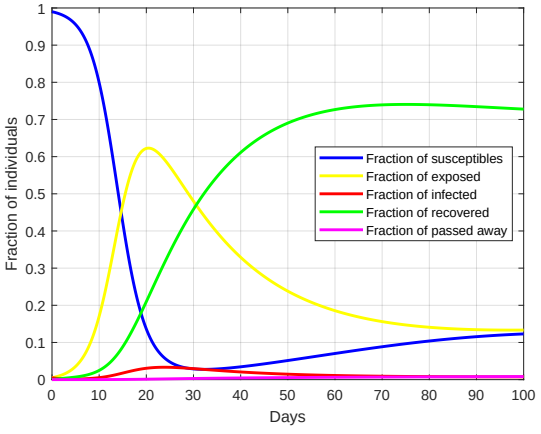
Under the assumption that the initial fraction of passed away individuals in transient state $p_\infty = 0$, this can be simplified to $R_0 = \frac{\alpha_e(\beta+\mu)+\kappa\alpha_i}{(\kappa+\rho)(\beta+\mu)}$. Similarly as in the SIR model, this reproduction number can be related to the stability results, that is, when $R_0 \leq 1$ the stability restriction of equilibrium point $\underline{y}_{\text{eq}}$ given by inequality (28) is satisfied. Similarly, when $R_0 > 1$ this equilibrium point is unstable. The above method can also be applied to the SIR model and can be found in section 6.1.8.

3.4 Numerical methods

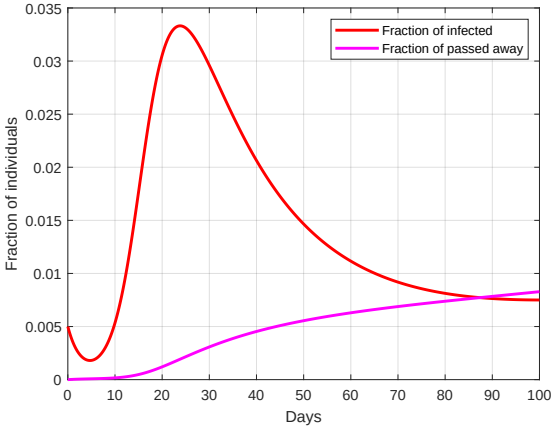
For numerical purposes, let

$$\alpha_e = 0.4,\ \alpha_i = 0.2,\ \beta = 0.35,\ \gamma = 0.01,\ \kappa = 0.02,\ \rho = 0.03,\ \mu = 0.005836, \tag{29}$$

where the mortality rate μ is assumed to have the same value as discussed in Section 2.6. Furthermore, let the initial condition be $\underline{y}(0) = (0.99, 0.005, 0.005, 0, 0)$, timestep $h = 0.01$ and a time period of 100 days. Then the (forward) Euler, the Runge-Kutta Dormand-Prince and the 4-stage Runge-Kutta method previously described yield the approximately same result given in Figures 18a and 18b. Similarly as for the SIR model, the population fractions of the individuals having the virus first increases rapidly and after that decreases, the fraction of susceptibles first decreases rapidly and after a while slowly increases and the recovered fraction starts increasing when the number of infected increases and steadily decreases after a while. The exposed fraction of the population is much higher than the infected fraction and the maximum exposed fraction of the population is at its peak on day 20, while the maximum fraction of infected has its peak shortly after that. Both these results rely on the fact that the infected population first belonged to the exposed category. Furthermore, the passed away fraction increases very slowly, due to the mortality rate only being $\mu = 0.005836$ and ends up to be less than one percent of the population after these first 100 days. Moreover, the stability restriction for equilibrium point $\underline{y}_{\text{eq}}$ given in (28) is not satisfied and hence this point is unstable.



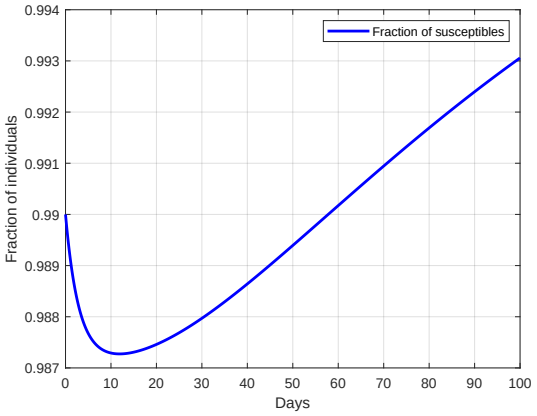
(a) The fraction of susceptibles, exposed, infected, recovered and passed away.



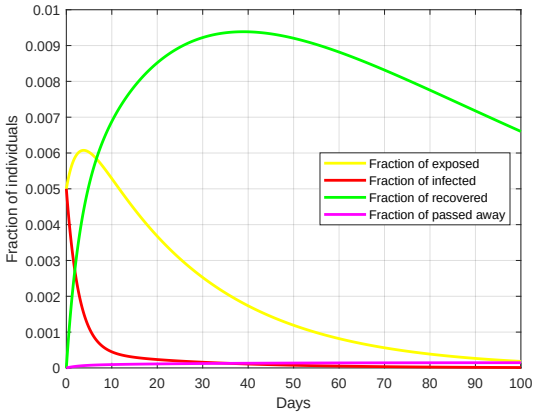
(b) The fraction of infected and passed away.

Figure 18: A distribution of the first 100 days in the SEIRP model with parameters $\alpha_e = 0.4$, $\alpha_i = 0.2$, $\beta = 0.35$, $\gamma = 0.01$, $\kappa = 0.02$, $\rho = 0.03$, $\mu = 0.005836$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.005, 0.005, 0, 0)^T$.

For numerical purposes, the value of α_e is changed to 0. In this case, the inequality given in equation (28) holds and hence $\underline{y}_{\text{eq}}$ is stable. This can clearly be seen in Figures 19a and 19b, where all parameters except for α_e remained the same, because the exposed, infected and recovered population converge to 0, while the fraction of susceptibles and fraction of passed away converge to some values $1 - p_\infty$ and p_∞, respectively. In Figure 19a it can be seen that the fraction of susceptibles decreases in the first ten days, but after that increases. In Figure 19b the fraction of exposed and recovered increases at first, but after that slowly decrease. On the other hand, the fraction of infected decreases immediately. Lastly, the passed away fraction increases slowly in the first days, but after that stabilizes.



(a) The fraction of susceptibles.



(b) The fraction of exposed, infected, recovered and passed away.

Figure 19: A distribution of the first 100 days in the SEIRP model with parameters $\alpha_e = 0$, $\alpha_i = 0.2$, $\beta = 0.35$, $\gamma = 0.01$, $\kappa = 0.02$, $\rho = 0.03$, $\mu = 0.005836$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.005, 0.005, 0, 0)^T$.

Finally, one could also add lockdowns and free days as done in the SIR model. This gives similar results as discussed in section 2.5 and can easily be visualized by using the Matlab code given in section 6.2.10.

3.5 The SEIRP model with vaccination

The model can be extended by introducing vaccinations. Assumed that susceptibles can be vaccinated with a rate θ ($0 \leq \theta \leq 1$), where they are transferred to the recovered compartment. Then by slightly changing system (24) one finds

$$\underline{y}' = \begin{cases} \dfrac{\mathrm{d}s(t)}{\mathrm{d}t} = -\alpha_e s(t)e(t) - \alpha_i s(t)i(t) + \gamma r(t) - \theta s(t), & \text{(30a)} \\ \dfrac{\mathrm{d}e(t)}{\mathrm{d}t} = \alpha_e s(t)e(t) + \alpha_i s(t)i(t) - \kappa e(t) - \rho e(t), & \text{(30b)} \\ \dfrac{\mathrm{d}i(t)}{\mathrm{d}t} = \kappa e(t) - \beta i(t) - \mu i(t), & \text{(30c)} \\ \dfrac{\mathrm{d}r(t)}{\mathrm{d}t} = \beta i(t) + \rho e(t) - \gamma r(t) + \theta s(t), & \text{(30d)} \\ \dfrac{\mathrm{d}p(t)}{\mathrm{d}t} = \mu i(t), & \text{(30e)} \end{cases}$$

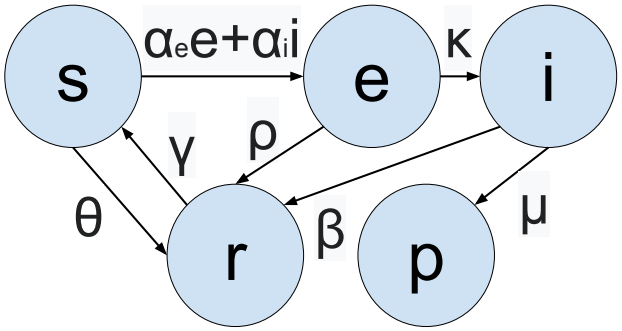
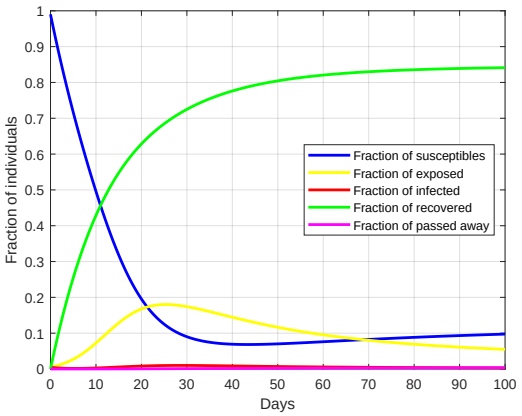
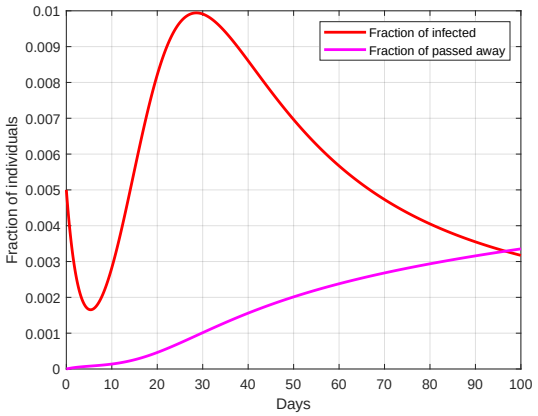


Figure 20: A flowchart of the SEIRP model with vaccinations.

for which the flowchart can be seen in Figure 20. For numerical purposes assume that $\theta = 0.06$ and the values of all other parameters are given in (29). Furthermore, let the initial condition be $\underline{y}(0) = (0.99, 0.005, 0.005, 0, 0)$, timestep $h = 0.01$ and a time period of 100 days. Using the (forward) Euler method one then finds Figures 21a and 21b. The population fractions in these figures behave similarly as in Figures 18a and 18b. However, the fraction of susceptibles, exposed, infected and passed away at each day is lower than in the latter two figures, while the fraction of recovered at each day is higher. These results are due to the extra flow from the susceptible to the recovered compartment.



(a) Fraction of susceptibles, exposed, infected, recovered and passed away.



(b) Fraction of infected and passed away.

Figure 21: A distribution of the first 100 days of the five compartments in the SEIRP model with vaccination and with parameters $\alpha_e = 0.4$, $\alpha_i = 0.2$, $\beta = 0.35$, $\gamma = 0.01$, $\kappa = 0.02$, $\rho = 0.03$, $\mu = 0.005836$, $\theta = 0.06$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.005, 0.005, 0, 0)^T$.

For better insight in the influence of the vaccination, assume now that the initial condition is changed to $\underline{y}(0) = (1, 0, 0, 0, 0)$. Hence, the only flow is between the susceptible and recovered compartments and there are no exposed, infected or passed away individuals in this model. Using these assumptions one finds with help of the (forward) Euler method Figure 22. Here, the fraction of recovered increases, while the fraction of susceptibles decreases. However, the rate at which the fraction of susceptibles become part of the recovered compartment decreases, which is due to lack of the fraction of available susceptibles. Furthermore, since the immunity loss parameter $\gamma = 0.01$ there is a flow going from the recovered to the susceptible compartment and hence the fraction of susceptibles does not converge to 0.

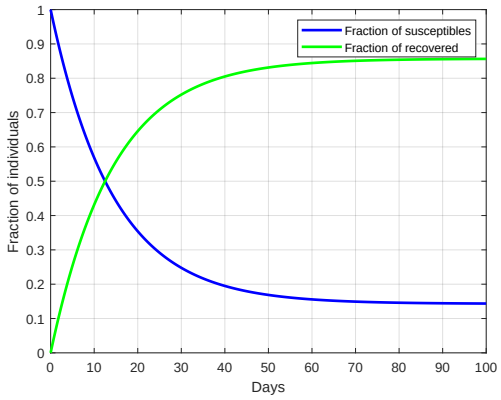


Figure 22: A distribution of the first 100 days of the susceptible and recovered compartments in the SEIRP model with vaccination with parameters $\alpha_e = 0.4$, $\alpha_i = 0.2$, $\beta = 0.35$, $\gamma = 0.01$, $\kappa = 0.02$, $\rho = 0.03$, $\mu = 0.005836$, $\theta = 0.06$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (1, 0, 0, 0, 0)^T$.

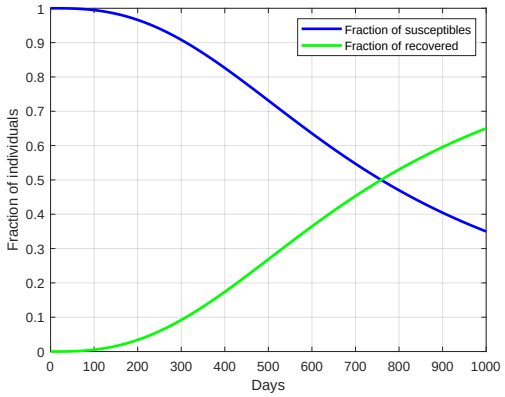


Figure 23: A distribution of the first 1000 days of the susceptible and recovered compartments in the SEIRP model with vaccination with parameters $\alpha_e = 0.4$, $\alpha_i = 0.2$, $\beta = 0.35$, $\gamma = 0.01$, $\kappa = 0.02$, $\rho = 0.03$, $\mu = 0.005836$, $\theta(t) = 2 \cdot 10^{-8} \cdot t^2 + 4 \cdot 10^{-10} \cdot t$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (1, 0, 0, 0, 0)^T$.

In reality, vaccinations are not available immediately. This can clearly be seen in Figure 24, where the coronavirus vaccine doses over time are given for different countries. Consequently, the vaccination rate θ should be modelled as a function over time. For numerical purposes, assume that

$$\theta(t) = \begin{cases} 2 \cdot 10^{-8} \cdot t^2 + 4 \cdot 10^{-10} \cdot t & \text{if } 2 \cdot 10^{-8} \cdot t^2 + 4 \cdot 10^{-10} \cdot t < 1, \\ 1 & \text{otherwise,} \end{cases} \tag{31}$$

where the time unit $t \geq 0$ is in days. Then, by using the same parameters and initial conditions as before, one finds Figure 23. Notice that the fraction of recovered behaves in a similar manner as the administered vaccine doses in Figure 24 [17], as desired.

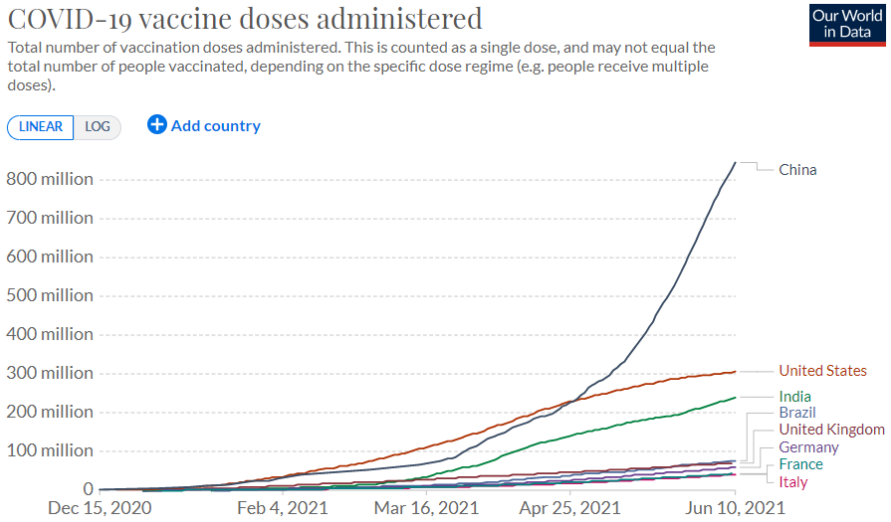
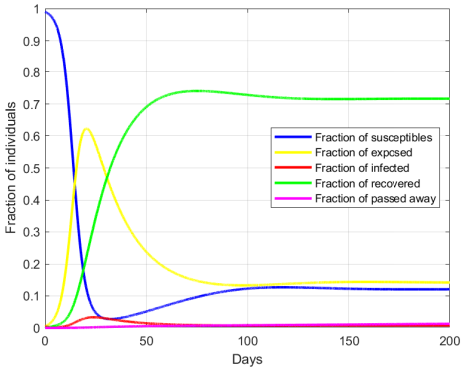
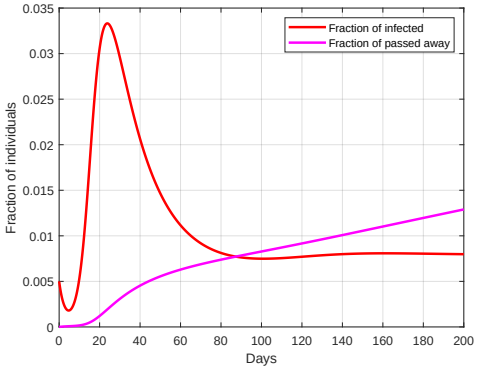


Figure 24: The coronavirus vaccine doses over time [17].

Now, let the parameters of system (30) be given by (29) and (31). Furthermore, let the initial condition be $\underline{y}(0) = (0.99, 0.005, 0.005, 0, 0)$, timestep $h = 0.01$ and a time period of 200 days. Using (forward) Euler, one then finds Figures 25a and 25b. Comparing the first days in these figures to the first days in figures 18a and 18b shows that the vaccinations do not influence the model much. This is due to the small value of θ in these first days. However, about day 200 it can be seen in Figure 25a that the fraction of recovered individuals is much higher than in Figure 18a, which shows the impact of vaccinations.



(a) The fraction of susceptibles, exposed, infected, recovered and passed away.



(b) The fraction of infected and passed away.

Figure 25: A distribution of the first 200 days in the SEIRP model with vaccination with parameters $\alpha_e = 0.4$, $\alpha_i = 0.2$, $\beta = 0.35$, $\gamma = 0.01$, $\kappa = 0.02$, $\rho = 0.03$, $\mu = 0.005836$, $\theta(t) = 2 \cdot 10^{-8} \cdot t^2 + 4 \cdot 10^{-10} \cdot t$, timestep $h = 0.01$ and initial condition $\underline{y}(0) = (0.99, 0.005, 0.005, 0, 0)^T$.

4 Conclusion and discussion

This study analyzed the SIR and SEIRP models to better understand the condition under which we can control the spread of viruses and the effectiveness of government interventions. For this reason, the total population was classified into different categories in both models. In the SIR model the maximum fraction of infected and the fraction of the population that ends up having the virus were calculated for different parameters. Depending on the specific circumstances and based on the findings of the calculations, governments can decide whether the coronavirus restrictions should be stricter or less strict to control the number of infected and how to manage the maximum capacity in hospitals. Both models calculated the equilibrium points, assessed the stability conditions of these equilibrium points and showed graphs to provide a clear overview of the distribution for each of the categories over time. In the SIR model a lockdown was introduced, which gives the utility of the coronavirus restrictions. Furthermore, by introducing a 'free days' period which was done in section 2.5 it could be seen that a decision to cancel all rules regarding the virus for a certain period is not a smart choice. In addition, the fatal SIR model discussed the impact of the virus in the long run with birth and death rates without vaccinations. In the SEIRP model a vaccination flow θ was added. When the coronavirus started spreading, the number of vaccinations were small, but increased over time and hence population distribution graphs were provided with $\theta(t)$ being a function over time, showing the impact of vaccinating the population.

Both mathematical models divided the total population into multiple categories. In particular, the susceptible, exposed, infected, recovered and passed away compartments were considered. However, in reality the infective compartment consists of smaller subcompartments such as the quarantined infected, symptomatic infected, hospitalized infected and super spreaders, which were not taken into account [18]. Furthermore, the total population can usually be divided into smaller subpopulations. Each of these subpopulations decides for themselves how strict they must adhere the coronavirus restrictions, which can cause fluctuations in the infection rates. These subpopulations are for instance schools, working places or different cultures. For example, in the beginning of 2021 thousands of Orthodox Christians ignored the coronavirus restrictions, due to worshipping their traditions [19]. Lastly, previous sections sometimes used parameters and functions such as α and θ which approximately represented reality, but in general these parameters are determined by making use of least squares methods. These methods make use of the curves representing the reality, then try to fit these curves and predict what the parameters of the system are [5][20].

Already before the coronavirus emerged, many disease models existed, but during the spreading new sophisticated models were discovered and existing models were thoroughly analyzed on high abstract levels. This study used mathematical models to predict the distribution of a pandemic over time and analyzed the effectiveness of interventions to control the spread of a virus. The findings stress the usefulness of mathematical modeling in better understanding and predicting conditions under which we can control the spread of the coronavirus. Future research could explore more types of interventions and make further distinctions in categories to refine the models. Overall, an advanced understanding is important, because increased insight may help to better control the spread of a virus on national and global scales. Moreover, it may also help regulators to assess the effectiveness of interventions and reduce the risks associated with the virus.

5 References

[1] (Mar. 2020). "Who announces covid-19 outbreak a pandemic," [Online]. Available: https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/news/news/2020/3/who-announces-covid-19-outbreak-a-pandemic.

[2] (Feb. 2021). "Lockdowns compared: Tracking governments' coronavirus responses," [Online]. Available: https://ig.ft.com/coronavirus-lockdowns/.

[3] (Feb. 2021). "UK finds more coronavirus cases with 'concerning' mutations," [Online]. Available: https://www.bbc.com/news/health-55900625.

[4] (Apr. 2019). "What is mathematical model," [Online]. Available: https://www.igi-global.com/dictionary/mathematical-model/18021.

[5] R. Sameni, "Mathematical modeling of epidemic diseases; a case study of the covid-19 coronavirus," *arXiv: Populations and Evolution*, 2020.

[6] D. B. M. William E. Boyce Richard C. Diprima, *Boyce's Elementary Differential Equations and Boundary Value Problems*, en. Wiley and Sons, 2017, ISBN: 978-1-119-38287-4.

[7] T. Kimura, "On dormand-prince method," p. 9, 2009.

[8] (Sep. 2019). "Total number of live births in the netherlands from 2008 to 2018," [Online]. Available: https://www.statista.com/statistics/519994/total-number-of-live-births-in-the-netherlands/.

[9] (Jan. 2021). "Antistoffen coronavirus zijn na acht maanden nog zichtbaar, en dat is goed nieuws voor de vaccins," [Online]. Available: https://www.trouw.nl/zorg/antistoffen-coronavirus-zijn-na-acht-maanden-nog-zichtbaar-en-dat-is-goed-nieuws-voor-de-vaccins~b06a11cd/.

[10] (May 2021). "Total coronavirus deaths in the netherlands," [Online]. Available: https://www.worldometers.info/coronavirus/country/netherlands/.

[11] (Dec. 2018). "Life expectancy netherlands," [Online]. Available: https://datatopics.worldbank.org/world-development-indicators/.

[12] (May 2021). "Questions and answers about the covid-19 vaccines," [Online]. Available: https://www.rivm.nl/en/covid-19-vaccination/questions-and-answers.

[13] W. O. Kermack and A. G. McKendrick, *A Contribution to the Mathematical Theory of Epidemics*. 1927.

[14] G. Shabbir, H. Khan, and M. Sadiq, "A note on exact solution of sir and sis epidemic models," *arXiv: Classical Analysis and ODEs*, 2010.

[15] P. van den Driessche and J. Watmough, "Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission," *Mathematical Biosciences*, p. 20, 2001.

[16] O. Diekmann, J. Heesterbeek, and M. Roberts, "The construction of next-generation matrices for compartmental epidemic models," *Journal of The Royal Society Interface*, vol. 7, pp. 873–885, 2009.

[17] (Jun. 2021). "Covid-19 vaccine doses administered," [Online]. Available: https://ourworldindata.org/grapher/cumulative-covid-vaccinations.

[18] M. Uddin, M. T. Nasseef, M. Mahmud, and A. AlArjani, "Application of mathematical modelings in prediction of covid-19 transmission dynamics," 2020.

[19] (Jan. 2021). "Some orthodox christians ignore covid warnings on epiphany," [Online]. Available: https://apnews.com/article/bulgaria-coronavirus-81e268b7bdb20dc126c9fe0f5feef928.

[20] A. Balcha, "Curve fitting and least square analysis to extrapolate for the case of covid-19 status in ethiopia," *Advances in Infectious Diseases*, vol. 10, pp. 143–159, 2020.

[21] J. ten Thije Boonkkamp, *Lecture notes: Theory and practice of ordinary differential equations (2WN30)*, en. 2020.

6 Appendices

6.1 Derivations

This section discusses some calculations and derivations which were previously used in the report.

6.1.1 Theorems

Theorem 1. *If the sum of each column of a square matrix equals* 0*, then* 0 *is one of the eigenvalues of the matrix.*

Proof. Let A be a $n \times n$ matrix for which each of the columns sums up to zero and let $\underline{v} \in \mathbb{R}^n$ be the all-ones vector. Then, one finds that $\underline{v}^T A = 0$ which implies that A is singular and hence has eigenvalue $\lambda = 0$. □

Theorem 2. *All Runge-Kutta methods preserve linear invariants.*

Proof. In general Runge-Kutta methods have the form

$$\underline{y}_{n+1} = \underline{y}_n + h \sum_{i=1}^{m} b_i \underline{k}_i,$$

where the b_i are the weights of the Runge-Kutta method and

$$\underline{k}_i = \underline{f}\Big(t_n + hc_i, \underline{y}_n + h \sum_{j=1}^{m} a_{ij} \underline{k}_j\Big),$$

for each $i = 1, 2, ..., m$, where c_i are the nodes of the Runge-Kutta method and a_{ij} are the entries of the Butcher array of the Runge-Kutta method. Recall that a variable $I(\underline{y})$ is an invariant of the autonomous ordinary differential equations system $\underline{y} = \underline{f}(\underline{y})$ if $I(\underline{y}(t))$ is a constant for all solutions of the system, i.e., for all $\underline{y}$ it holds that

$$\frac{\mathrm{d}}{\mathrm{d}t} I(\underline{y}) = \nabla I(\underline{y})^T \underline{y}' = \nabla I(\underline{y})^T \underline{f}(\underline{y}) = 0.$$

If we let $I(\underline{y}) = \underline{a}^T \underline{y}$ be a linear invariant, then

$$\frac{\mathrm{d}}{\mathrm{d}t} I(\underline{y}) = \nabla I(\underline{y})^T \underline{f}(\underline{y}) = \underline{a}^T \underline{f}(\underline{y}) = 0,$$

for all values of $\underline{y}$ and hence $\underline{a}^T \underline{k}_i = 0$ for all $i = 1, 2, ..., m$ which yields

$$I_{n+1} = \underline{a}^T \underline{y}_{n+1} = \underline{a}^T \underline{y}_n + h \sum_{i=1}^{m} b_i \underline{a}^T \underline{k}_i = I_n.$$

Consequently, I_n is a constant for all values of n and is hence an invariant of the Runge-Kutta methods [21]. □

6.1.2 Jacobian in the SIR model

The characteristic equation given by (15) can be rewritten as follows:

$$
\begin{aligned}
&(-\alpha^2 y_2 y_1 + \alpha\beta y_2 + \alpha y_2\lambda - \alpha y_1\lambda + \beta\lambda + \lambda^2)(-\gamma - \lambda) - \alpha y_2(\alpha\gamma y_1 + \alpha y_1\lambda - \gamma\beta) = 0,\\
\implies &\alpha^2\gamma y_1 y_2 - \alpha\beta\gamma y_2 - \alpha\gamma y_2\lambda + \alpha\gamma y_1\lambda - \beta\gamma\lambda - \gamma\lambda^2 + \alpha^2 y_2 y_1\lambda - \alpha\beta y_2\lambda - \alpha y_2\lambda^2 +\\
&\alpha y_1\lambda^2 - \beta\lambda^2 - \lambda^3 - \alpha^2\gamma y_1 y_2 - \alpha^2 y_1 y_2\lambda + \alpha\beta\gamma y_2 = 0,\\
\implies &-\alpha\gamma y_2\lambda + \alpha\gamma y_1\lambda - \beta\gamma\lambda - \gamma\lambda^2 - \alpha\beta y_2\lambda - \alpha y_2\lambda^2 + \alpha y_1\lambda^2 - \beta\lambda^2 - \lambda^3 = 0,\\
\implies &\alpha\gamma y_2\lambda - \alpha\gamma y_1\lambda + \beta\gamma\lambda + \gamma\lambda^2 + \alpha\beta y_2\lambda + \alpha y_2\lambda^2 - \alpha y_1\lambda^2 + \beta\lambda^2 + \lambda^3 = 0,\\
\implies &\lambda^3 + (\gamma + \alpha y_2 - \alpha y_1 + \beta)\lambda^2 + (\alpha\gamma y_2 - \alpha\gamma y_1 + \beta\gamma + \alpha\beta y_2)\lambda = 0,\\
\implies &\lambda(\lambda^2 + (\gamma + \alpha y_2 - \alpha y_1 + \beta)\lambda + \alpha\gamma y_2 - \alpha\gamma y_1 + \beta\gamma + \alpha\beta y_2)) = 0,\\
\implies &\lambda = 0 \text{ or } \lambda = \frac{-\gamma - \alpha y_2 + \alpha y_1 - \beta \pm \sqrt{(\gamma + \alpha y_2 - \alpha y_1 + \beta)^2 + 4\alpha\gamma y_1 - 4(\alpha\gamma y_2 + \beta\gamma + \alpha\beta y_2)}}{2}.
\end{aligned}
$$

6.1.3 SIR Eigenvalues for equilibrium point 1

For the first equilibrium point $\underline{y}_{\text{eq1}}$ one finds the eigenvalues $\lambda = 0$ and

$$
\begin{aligned}
\lambda &= \frac{-\gamma + \alpha - \beta \pm \sqrt{(\gamma - \alpha + \beta)^2 + 4\alpha\gamma - 4(\beta\gamma)}}{2},\\
&= \frac{\alpha - \beta - \gamma \pm \sqrt{\alpha^2 + \gamma^2 + \beta^2 - 2\alpha\beta - 2\alpha\gamma + 2\beta\gamma + 4\alpha\gamma - 4\beta\gamma}}{2},\\
&= \frac{\alpha - \beta - \gamma \pm \sqrt{\alpha^2 + \beta^2 + \gamma^2 + 2\alpha\gamma - 2(\alpha\beta + \beta\gamma)}}{2},\\
&= \frac{\alpha - \beta - \gamma \pm \sqrt{(\alpha - \beta + \gamma)^2}}{2} = \frac{\alpha - \beta - \gamma \pm |\alpha - \beta + \gamma|}{2}.
\end{aligned}
$$

For the first of these two eigenvalues:

$$
\lambda = \begin{cases} \frac{\alpha-\beta-\gamma+\alpha-\beta+\gamma}{2} = \frac{2\alpha-2\beta}{2} = \alpha - \beta & \text{if } \alpha - \beta + \gamma \geq 0, \\ \frac{\alpha-\beta-\gamma-\alpha+\beta-\gamma}{2} = -\gamma & \text{if } \alpha - \beta + \gamma < 0, \end{cases}
$$

and for the second:

$$
\lambda = \begin{cases} \frac{\alpha-\beta-\gamma+\alpha-\beta+\gamma}{2} = \frac{2\alpha-2\beta}{2} = \alpha - \beta & \text{if } \alpha - \beta + \gamma < 0, \\ \frac{\alpha-\beta-\gamma-\alpha+\beta-\gamma}{2} = -\gamma & \text{if } \alpha - \beta + \gamma \geq 0. \end{cases}
$$

6.1.4 SIR Eigenvalues for equilibrium point 2

For the second equilibrium point $\underline{y}_{\text{eq2}}$ one finds $\lambda = 0$ and

$$\begin{aligned}
\lambda &= \frac{-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} \pm \sqrt{(\gamma + \frac{\gamma(\alpha-\beta)}{\gamma+\beta})^2 + 4\beta\gamma - 4(\frac{\gamma^2(\alpha-\beta)}{\gamma+\beta} + \beta\gamma + \frac{\beta\gamma(\alpha-\beta)}{\gamma+\beta})}}{2}, \\
&= \frac{-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} \pm \sqrt{\gamma^2 + 2\frac{\gamma^2(\alpha-\beta)}{\gamma+\beta} + \frac{\gamma^2(\alpha-\beta)^2}{(\gamma+\beta)^2} + 4\beta\gamma - 4\frac{\gamma^2(\alpha-\beta)}{\gamma+\beta} - 4\beta\gamma - 4\frac{\beta\gamma(\alpha-\beta)}{\gamma+\beta}}}{2}, \\
&= \frac{-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} \pm \sqrt{\gamma^2 + \frac{\gamma^2(\alpha-\beta)^2}{(\gamma+\beta)^2} - 2\frac{\gamma^2(\alpha-\beta)}{\gamma+\beta} - 4\frac{\beta\gamma(\alpha-\beta)}{\gamma+\beta}}}{2}, \\
&= \frac{-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} \pm \sqrt{\gamma^2 + \frac{\gamma^2}{(\gamma+\beta)^2} \cdot (\alpha^2 + \beta^2 - 2\alpha\beta) - \frac{2\gamma^2}{\gamma+\beta} \cdot (\alpha - \beta) - \frac{4\beta\gamma}{\gamma+\beta} \cdot (\alpha - \beta)}}{2}, \\
&= \frac{-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} \pm \sqrt{\frac{\gamma^2}{(\gamma+\beta)^2}\alpha^2 + \left(-\frac{2\beta\gamma^2}{(\gamma+\beta)^2} - \frac{2\gamma^2}{(\gamma+\beta)} - \frac{4\beta\gamma}{\gamma+\beta}\right)\alpha + \left(\gamma^2 + \frac{\gamma^2\beta^2}{(\gamma+\beta)^2} + \frac{2\beta\gamma^2}{\gamma+\beta} + \frac{4\beta^2\gamma}{\gamma+\beta}\right)}}{2}.
\end{aligned}$$

These last two eigenvalues are complex if the part inside the square root is negative, which can only occur if $\gamma > 0$. Using Mathematica one then finds that this is the case when

$$\frac{2\beta^2 + 4\beta\gamma + \gamma^2 - 2\sqrt{\beta^4 + 3\beta^3\gamma + 3\beta^2\gamma^2 + \beta\gamma^3}}{\gamma} < \alpha < \frac{2\beta^2 + 4\beta\gamma + \gamma^2 + 2\sqrt{\beta^4 + 3\beta^3\gamma + 3\beta^2\gamma^2 + \beta\gamma^3}}{\gamma}, \tag{32}$$

for which a visual representation is given in Figure 26. Furthermore, if these last two eigenvalues are complex, then the real part of these eigenvalues equals zero when

$$-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} = 0 \implies -(\gamma+\beta) = (\alpha-\beta) \implies \alpha = -\gamma.$$

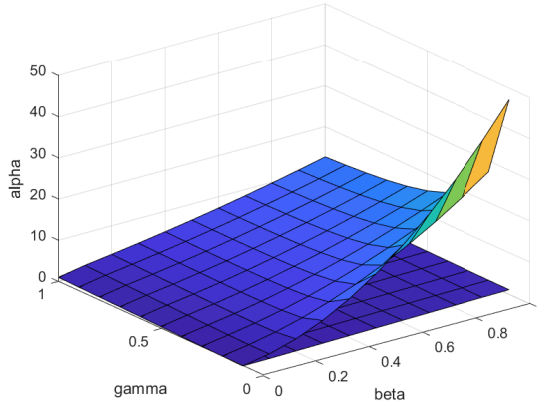


Figure 26: A 3D plot of all values satisfying equation (32)

Consequently, since it was assumed that $\alpha, \gamma > 0$ this tells us that the real part of the eigenvalues are always negative and hence stable if the eigenvalues are complex [6]. However, if all three eigenvalues are real, one might be interested in when this second equilibrium point is stable. This is the case when all three eigenvalues are not positive. Choosing the eigenvalue with the negative sign above gives for all positive parameters α, β and γ a negative eigenvalue. Therefore, next it will be determined when the other eigenvalue with the positive sign above is non-positive. It follows

that

$$\frac{-\gamma - \frac{\gamma(\alpha-\beta)}{\gamma+\beta} + \sqrt{\frac{\gamma^2}{(\gamma+\beta)^2}\alpha^2 + \left(-\frac{2\beta\gamma^2}{(\gamma+\beta)^2} - \frac{2\gamma^2}{(\gamma+\beta)} - \frac{4\beta\gamma}{\gamma+\beta}\right)\alpha + \left(\gamma^2 + \frac{\gamma^2\beta^2}{(\gamma+\beta)^2} + \frac{2\beta\gamma^2}{\gamma+\beta} + \frac{4\beta^2\gamma}{\gamma+\beta}\right)}}{2} \leq 0,$$

$$\implies \sqrt{\frac{\gamma^2}{(\gamma+\beta)^2}\alpha^2 + \left(-\frac{2\beta\gamma^2}{(\gamma+\beta)^2} - \frac{2\gamma^2}{(\gamma+\beta)} - \frac{4\beta\gamma}{\gamma+\beta}\right)\alpha + \left(\gamma^2 + \frac{\gamma^2\beta^2}{(\gamma+\beta)^2} + \frac{2\beta\gamma^2}{\gamma+\beta} + \frac{4\beta^2\gamma}{\gamma+\beta}\right)} \leq \gamma + \frac{\gamma(\alpha-\beta)}{\gamma+\beta},$$

$$\implies \frac{\gamma^2}{(\gamma+\beta)^2}\alpha^2 + \left(-\frac{2\beta\gamma^2}{(\gamma+\beta)^2} - \frac{2\gamma^2}{(\gamma+\beta)} - \frac{4\beta\gamma}{\gamma+\beta}\right)\alpha + \left(\gamma^2 + \frac{\gamma^2\beta^2}{(\gamma+\beta)^2} + \frac{2\beta\gamma^2}{\gamma+\beta} + \frac{4\beta^2\gamma}{\gamma+\beta}\right) \leq \gamma^2 + \frac{\gamma^2(\alpha-\beta)^2}{(\gamma+\beta)^2} + 2\frac{\gamma^2(\alpha-\beta)}{(\gamma+\beta)} = \gamma^2 + \frac{\gamma^2\alpha^2}{(\gamma+\beta)^2} + \frac{\gamma^2\beta^2}{(\gamma+\beta)^2} - 2\frac{\gamma^2\alpha\beta}{(\gamma+\beta)^2} + 2\frac{\gamma^2\alpha}{\gamma+\beta} - 2\frac{\gamma^2\beta}{\gamma+\beta},$$

$$\implies \left(-\frac{2\gamma^2}{(\gamma+\beta)} - \frac{4\beta\gamma}{\gamma+\beta}\right)\alpha + \left(\frac{2\beta\gamma^2}{\gamma+\beta} + \frac{4\beta^2\gamma}{\gamma+\beta}\right) \leq 2\frac{\gamma^2\alpha}{\gamma+\beta} - 2\frac{\gamma^2\beta}{\gamma+\beta},$$

$$\implies \left(-\frac{4\gamma^2}{(\gamma+\beta)} - \frac{4\beta\gamma}{\gamma+\beta}\right)\alpha + \left(\frac{4\beta\gamma^2}{\gamma+\beta} + \frac{4\beta^2\gamma}{\gamma+\beta}\right) \leq 0,$$

$$\implies \alpha \geq \frac{\beta\gamma^2 + \beta^2\gamma}{\gamma^2 + \beta\gamma} = \frac{\beta\gamma + \beta^2}{\gamma+\beta} = \beta,$$

and therefore the second equilibrium point is stable when $\alpha \geq \beta$.

6.1.5 Eigenvalues SEIRP

The characteristic equation of the Jacobian matrix given in equation (26) is

$$\begin{aligned}
&-\lambda \cdot \begin{vmatrix} \alpha_e(1-p_\infty) - \kappa - \rho - \lambda & \alpha_i(1-p_\infty) & 0 & 0 \\ \kappa & -\beta-\mu-\lambda & 0 & 0 \\ \rho & \beta & -\gamma-\lambda & 0 \\ 0 & \mu & 0 & -\lambda \end{vmatrix} \\
&= -\lambda\Bigg((\alpha_e(1-p_\infty) - \kappa - \rho - \lambda) \cdot \begin{vmatrix} -\beta-\mu-\lambda & 0 & 0 \\ \beta & -\gamma-\lambda & 0 \\ \mu & 0 & -\lambda \end{vmatrix} - \kappa \begin{vmatrix} \alpha_i(1-p_\infty) & 0 & 0 \\ \beta & -\gamma-\lambda & 0 \\ \mu & 0 & -\lambda \end{vmatrix} + \\
&\quad \rho \begin{vmatrix} \alpha_i(1-p_\infty) & 0 & 0 \\ -\beta-\mu-\lambda & 0 & 0 \\ \mu & 0 & -\lambda \end{vmatrix} \Bigg) \\
&= -\lambda\Big((\alpha_e(1-p_\infty) - \kappa - \rho - \lambda)(-\beta-\mu-\lambda)(-\gamma-\lambda)(-\lambda) - \kappa(\alpha_i(1-p_\infty))(-\gamma-\lambda)(-\lambda) \Big) \\
&= -\lambda^2(\lambda+\gamma)\Big((\alpha_e(1-p_\infty) - \kappa - \rho - \lambda)(-\beta-\mu-\lambda) - \kappa(\alpha_i(1-p_\infty)) \Big) \\
&= -\lambda^2(\lambda+\gamma)\Big(-\beta(\alpha_e(1-p_\infty) - \kappa - \rho - \lambda) - \mu(\alpha_e(1-p_\infty) - \kappa - \rho - \lambda) - \\
&\quad \lambda(\alpha_e(1-p_\infty) - \kappa - \rho) + \lambda^2 - \kappa\alpha_i(1-p_\infty) \Big) \\
&= -\lambda^2(\lambda+\gamma)\left(\lambda^2 + \lambda(\beta+\mu+\alpha_e(1-p_\infty) - \kappa - \rho) - (\alpha_e(1-p_\infty) - \kappa - \rho)(\beta+\mu) - \kappa\alpha_i\right) = 0,
\end{aligned}$$

resulting in eigenvalues

$$\lambda_1 = \lambda_2 = 0,\ \lambda_3 = -\gamma,\ \lambda_{4,5} = \frac{\alpha_e(1-p_\infty) - \kappa - \rho - \beta - \mu \pm \sqrt{(\beta+\mu+\alpha_e(1-p_\infty) - \kappa - \rho)^2 + 4\kappa\alpha_i}}{2}.$$

6.1.6 Largest eigenvalue SEIRP

The largest eigenvalue λ_4 given in equation (27) is less or equal to 0 when:

$$\begin{aligned}
\lambda_4 \leq 0 &\implies \frac{\alpha_e(1-p_\infty)-\kappa-\rho-\beta-\mu+\sqrt{(\beta+\mu+\alpha_e(1-p_\infty)-\kappa-\rho)^2+4\kappa\alpha_i}}{2} \leq 0,\\
&\implies \sqrt{(\beta+\mu+\alpha_e(1-p_\infty)-\kappa-\rho)^2+4\kappa\alpha_i} \leq \beta+\mu-(\alpha_e(1-p_\infty)-\kappa-\rho),\\
&\implies (\beta+\mu+\alpha_e(1-p_\infty)-\kappa-\rho)^2+4\kappa\alpha_i \leq (\alpha_e(1-p_\infty)-\kappa-\rho)^2+\beta^2+\mu^2+\\
&\quad 2\beta\mu-2\beta(\alpha_e(1-p_\infty)-\kappa-\rho)-2\mu(\alpha_e(1-p_\infty)-\kappa-\rho),\\
&\implies (\alpha_e(1-p_\infty)-\kappa-\rho)^2+\beta^2+\mu^2+2\beta(\alpha_e(1-p_\infty)-\kappa-\rho)+2\mu(\alpha_e(1-p_\infty)-\kappa-\rho)+\\
&\quad 2\beta\mu+4\kappa\alpha_i \leq (\alpha_e(1-p_\infty)-\kappa-\rho)^2+\beta^2+\mu^2+2\beta\mu-2\beta(\alpha_e(1-p_\infty)-\kappa-\rho)-\\
&\quad 2\mu(\alpha_e(1-p_\infty)-\kappa-\rho),\\
&\implies \beta(\alpha_e(1-p_\infty)-\kappa-\rho)+\mu(\alpha_e(1-p_\infty)-\kappa-\rho)+\kappa\alpha_i \leq 0,\\
&\implies \kappa\alpha_i \leq (\kappa+\rho-\alpha_e)(\beta+\mu).
\end{aligned}$$

6.1.7 Conditions disease-free equilibrium

The disease-free equilibrium point given in equation (25) satisfies each of the five conditions described in section 2 of [15]. The first of these conditions is that if $\underline{y}_{\text{eq}} \geq \underline{0}$, then each of the entries of $\mathcal{F}$, $\mathcal{V}_i^+$ and $\mathcal{V}_i^-$ is non-negative. The second condition states that if the i-th entry of $\underline{y}_{\text{eq}}$ is equal to 0, then the same entry in $\mathcal{V}_i^-$ is equal to 0. The third condition tells us that $\mathcal{F}_i = 0$ for all disease-free compartments i. If $\underline{y}_{\text{eq}}$ is a disease-free equilibrium point, the fourth condition states that $\mathcal{F}_i(\underline{y}_{\text{eq}}) = 0$ and $\mathcal{V}_i^+(\underline{y}_{\text{eq}}) = 0$ for all exposed and infected compartments. The fifth condition is that if $\mathcal{F}$ is set to 0, the eigenvalues of the Jacobian of the system $\underline{f}(\underline{y}_{\text{eq}})$ are non-positive, i.e. the equilibrium point is stable.

6.1.8 Reproduction number SIR model

The reproduction number in the SIR model can in a similar way be constructed as in the SEIRP model by using the next generation matrix. The set of equations $\underline{f}(\underline{y})$ given in (4) will be written in the form $\underline{f}(\underline{y}) = \underline{\mathcal{F}}(\underline{y}) + \underline{\mathcal{V}}(\underline{y})$, where $\underline{\mathcal{F}}$ consists of all new infection terms of $\underline{f}(\underline{y})$ and $\underline{\mathcal{V}}(\underline{y})$ of the remaining terms. Consequently, one finds

$$\underline{\mathcal{F}} = \begin{pmatrix} 0 \\ \alpha s(t) i(t) \\ 0 \end{pmatrix} \text{ and } \underline{\mathcal{V}} = \begin{pmatrix} -\alpha s(t) i(t) + \gamma r(t) \\ -\beta i(t) \\ \beta i(t) - \gamma r(t) \end{pmatrix}.$$

The only disease-free equilibrium point of the SIR model given in equation (13) satisfies each of the five conditions described [15] and section 6.1.7 above. Consequently, one can make use of Lemma 1 in section 3.3 and in this article. Using this lemma on the disease-free equilibrium point (13) yields Jacobians

$$\nabla_{\underline{y}} \underline{\mathcal{F}}(\underline{y}_{\text{eq1}}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ and } \nabla_{\underline{y}} \underline{\mathcal{V}}(\underline{y}_{\text{eq1}}) = \begin{pmatrix} 0 & -\alpha & \gamma \\ 0 & -\beta & 0 \\ 0 & \beta & -\gamma \end{pmatrix}.$$

Notice that $\nabla_{\underline{y}} \underline{\mathcal{F}}(\underline{y}_{\text{eq1}}) + \nabla_{\underline{y}} \underline{\mathcal{V}}(\underline{y}_{\text{eq1}}) = J(\underline{y}_{\text{eq1}})$ where $J(\underline{y})$ is the Jacobian given in equation (15), as desired. Consequently, one finds

$$F(\underline{y}_{\text{eq1}}) = \begin{pmatrix} \alpha \end{pmatrix} \text{ and } V(\underline{y}_{\text{eq1}}) = \begin{pmatrix} -\beta \end{pmatrix}.$$

Then, similarly as in the SEIRP model, one can make use of Theorem 2 in [15], which gives $R_0 = \rho(-FV^{-1}) = \frac{\alpha}{\beta}$. This agrees with the reproduction found in section (2.2).

6.2 Matlab code

This section consists of Matlab code used to make graphs and compute numerical results. In each code the parameters can easily be modified to give numerical results.

6.2.1 Newton-Raphson method

The Matlab code below is used to plot Figures 2 and 3. Furthermore, it applies the Newton-Rhapson method which yields Figure 4.

```
%% Newton-Raphson method
clear all; clc; tic % tic measures the time

%{
% Plot of i_max(R_0)
i0 = 0.01; s0 = 0.99; xbegin = 1; xend = 9; ybegin = 0; yend = 1;
fplot(@(x) i0+s0-((1/x)*(1+log(x*s0))),[xbegin xend])
ylim([ybegin yend]); xlabel('R_0'); ylabel('i$_{\mathrm{max}}$','
    Interpreter','latex'); saveas(gcf,'Function_i_max.pdf'); % saving a
     pdf of the plot
%}

%{
% Plot of f(s(infinity),R)
R_1=0.0; R_2=10; R_3=4; R_4=2; R_5=1.4; R_6=1.; R_7=0.8; xbegin = 0.98;
     xend = 1; ybegin = -0.02; yend = 0.005;
fplot(@(x) x+(1/R_2)*log(0.99./x)-1,[xbegin xend])
ylim([ybegin yend]); hold on;
fplot(@(x) x+(1/R_3)*log(0.99./x)-1,[xbegin xend]);
fplot(@(x) x+(1/R_4)*log(0.99./x)-1,[xbegin xend]);
fplot(@(x) x+(1/R_5)*log(0.99./x)-1,[xbegin xend]);
fplot(@(x) x+(1/R_6)*log(0.99./x)-1,[xbegin xend]);
fplot(@(x) x+(1/R_7)*log(0.99./x)-1,[xbegin xend]);
fplot(@(x) 0*x,[xbegin xend],'k');
legend({'R_0=10','R_0=4','R_0=2','R_0=1.2','R_0=1','R_0=0.8'}, '
    Location','northwest'); xlabel('x'); ylabel('f(x)'); hold off;
    saveas(gcf,'Zeros_C_values2.pdf');
%}

%{
%Newton-Raphson iteration
r = zeros(1,500); s = zeros(1,500);
for j = 1:1:500 %R=0.01,0.02,...,2.00
R=100/j; x = zeros(1, 10); x(1)=1*10^(-100); % initial condition
for i = 2:1:100000 % 100000/100=1000 iterations
x(i)=x(i-1)-(x(i-1)+R*log(0.99/(x(i-1)))-1)/(1-R/(x(i-1)));
end
s(j) = x(100); % the value of s for R=0.01,0.02,...,2.00
r(j)=1-s(j); % the value of r for R=0.01,0.02,...,2.00
end
x = linspace(0.01,5,500); x = x.'; r = r.'; plot(x, r); xlabel('R_0');
    ylabel('Infected fraction r(t_\infty)'); title('Total fraction of
    population infected when t to infinity'); saveas(gcf,'
    NewtonRaphson1.pdf');
%}
toc % measures the time
```

6.2.2 Plot of complex eigenvalues in the SIR model

The Matlab code below is used to plot the graph given in Figure 26.

```
%% Alpha values 3D plot
clear; close all; clc;
x = 0.0:0.1:1; y = 0.0:0.1:1; [X,Y] = meshgrid(x,y);
Z = (((2.*((X).^2))+4.*X.*Y+(Y).^2)./Y)+((2.*sqrt(X.^4+3.*((X).^3).*Y
    +3.*((X).^2).*((Y).^2)+X.*((Y).^3)))./Y);
surf(X,Y,Z); hold on;
Z = (((2.*((X).^2))+4.*X.*Y+(Y).^2)./Y)-((2.*sqrt(X.^4+3.*((X).^3).*Y
    +3.*((X).^2).*((Y).^2)+X.*((Y).^3)))./Y);
surf(X,Y,Z); xlabel('beta'); ylabel('gamma'); zlabel('alpha'); saveas(
    gcf,'3D_plot.pdf'); hold off
```

6.2.3 (Forward) Euler method on the SIR model.

The Matlab code below yields a graph of the compartment division in the SIR model by using the (forward) Euler method.

```
%% (Forward) Euler method for the SIR model
clear; close all; clc; format long;

% Parameters
alpha = 0.15; beta = 0.15; gamma = 0.01; s0 = 1-(10^(-2)); i0 =
    (10^(-2)); r0 = 0.0; T = 1000; dt = 0.01; % length of the time
    interval (dt should hence divide T)
fprintf('The value of the reproduction number R0 is %.2f. ',alpha/beta)

% Calculation of the SIR values over time
[s,i,r] = sir1(alpha,beta,gamma,s0,i0,r0,T,dt);

% Plot of the population division over time
t = 0:dt:T-dt; plot(t,s,'b',t,i,'r',t,r,'g','LineWidth',2); grid on;
    xlabel('Days'); ylabel('Fraction of individuals'); title('
    Population fractions in the basic SIR model over time'); legend({'
    Fraction of susceptibles','Fraction of infected','Fraction of
    recovered'}, 'Location', 'northeast'); saveas(gcf,'Euler_SIR_test.
    pdf');
```

In this code the function SIR1 is used, which is given below.

```
function [s,i,r] = sir1(alpha,beta,gamma,s0,i0,r0,T,dt)
    s = zeros(1,T/dt); i = s; r=s; s(1) = s0; i(1) = i0; r(1) = r0;
    for t = 1:(T/dt)-1 % for each timestep t determining s(t), i(t) and
        r(t)
        s(t+1) = (-alpha * s(t) * i(t) + gamma * r(t)) * dt + s(t);
        i(t+1) = (alpha * s(t) * i(t) - beta * i(t))* dt + i(t);
        r(t+1) = (beta * i(t) - gamma * r(t)) * dt + r(t);
    end
end
```

6.2.4 Prince Dormand method on the SIR model.

The Matlab code below yields a graph of the compartment division in the SIR model by using the Dormand Prince method. In Matlab this method has a built in function called ODE45. Furthermore, this code computes 2d and 3d phase plots and 2d and 3d direction fields.

```
%% Runge–Kutta method for the SIR model
clear all; clc;

%Parameters & Initial conditions
alpha = 0.4; beta = 0.06; gamma = 0.01; S0 = 1-(10^(-2)); I0 = (10^(-2)); R0 = 0;

% Calculations of the SIR values over time
t0 = 0; tmax = 100; X0 = [S0; I0; R0]; options = odeset('RelTol',1e-6,'AbsTol',1e-6);
[ T,X ] = ode45( @(T,X) sir2(T,X,alpha,beta,gamma), [t0 tmax], X0, options );
%{
% A plot of S, I and R
plot(T,X(:,1),'b',T,X(:,2),'r',T,X(:,3),'g'); grid on; xlabel('Days'); ylabel('Fraction of individuals'); title('Population fractions in the basic SIR model over time (Runge–Kutta)'); legend('Fraction of susceptibles','Fraction of infected','Fraction of recovered'); saveas(gcf,'PopulationstestSIR2.pdf');
%}
%{
%A phase plot of S, I and R
plot3(X(:,1),X(:,2),X(:,3),'–o'); grid on; xlabel('s'); ylabel('i'); zlabel('r'); title('Phase plane plot of S versus I'); saveas(gcf,'PhasePlotSIR.pdf'); % saving a pdf of the plot
%}
% 3d Direction field plot of S, I, R.
t=0:.20:1; y=0:.20:1; z=0:.20:1; [T,Y,Z]=meshgrid(t,y,z);
dT= -alpha*T.*Y + gamma*Z; dY= alpha*T.*Y - beta*Y; dZ = beta*Y - gamma*Z; % new variables: T=S, Y=I and Z=R
N=sqrt(dT.^2+dY.^2+dZ.^2); dT=dT./N; dY=dY./N; dZ=dZ./N; quiver3(T,Y,Z,dT,dY,dZ); axis equal; axis([-0.02 1 -0.02 1]); grid on; xlabel('S'); ylabel('I'); zlabel('R'); title('A 3D plot of the direction field'); hold on;
plot3(1,0,0,'o','color','red'); plot3(beta/alpha,gamma*(alpha-beta)/(alpha*(gamma+beta)),beta*(alpha-beta)/(alpha*(gamma+beta)),'o','color','red'); %plot equilibrium points
hold off
%}

%{
% 2d Direction field plot
t=0:.05:1; y=0:.05:1; z=0.70:.10:0.70; % z fixed
%t=0.7:.05:0.7; y=0:.05:1; z=0:.05:1; % t fixed
%t=0:.05:1; y=0.7:.05:0.7; z=0:.05:1; % y fixed
[T,Y,Z]=meshgrid(t,y,z);
dT= -alpha*T.*Y + gamma*Z; dY= alpha*T.*Y - beta*Y; dZ = beta*Y - gamma*Z; % new variables: T=S, Y=I and Z=R
N=sqrt(dT.^2+dY.^2+dZ.^2); dT=dT./N; dY=dY./N; dZ=dZ./N;
quiver(T,Y,dT,dY) % z fixed
%quiver(Y,Z,dY,dZ) % t fixed
%quiver(T,Z,dT,dZ) % y fixed
axis equal; axis([-0.02 1 -0.02 1]); grid on; xlabel('s'); ylabel('i'); %zlabel('r');
startx = [0 0.6 0.95 1]; starty = [0.05 0.02 0.02 0.35]; streamline0 = streamline(T,Y,dT,dY,startx,starty); set(streamline0, 'LineWidth'
```

```
    ,1,'Color', '[1 0.5 0]')
%title('A plot of the direction field')
%plot equilibrium points
hold on
%plot(1,0,'MarkerSize', 0.4);
plot(1,0,'ro','color','red','MarkerFaceColor', 'r'); plot(beta/alpha,
    gamma*(alpha-beta)/(alpha*(gamma+beta)),'o','color','red','
    MarkerFaceColor', 'r');
saveas(gcf,'Direction_field_SIR.pdf'); % saving a pdf of the plot
hold off
%}
```

In this code the function SIR2 is used, which is given below.

```
function dx2 = sir2(T,X,alpha,beta,gamma)
    dx2 = zeros(3,1);
    dx2(1) = -alpha*X(1).*X(2) + gamma*X(3);
    dx2(2) = alpha*X(1).*X(2) - beta*X(2);
    dx2(3) = beta*X(2) - gamma*X(3);
end
```

6.2.5 4-stage Runge-Kutta method on the SIR model

The Matlab code below yields a graph of the compartment division in the SIR model by using the 4-stage Runge-Kutta method. Additionally, lockdowns and free days can be computed in this code by setting the variable(s) LOCKDOWN_PERIOD and FREE_PERIOD unequal to 0.

```
%% 4-stage RK method for the SIR model
clear; close all; clc;

% Parameters
alpha = 0.15; beta = 0.125; gamma = 0.01; s0 = 1-(10^(-2)); i0 =
    (10^(-2)); r0 = 0.0; T = 100; dt = 0.01;
fprintf('The value of the reproduction number R0 (without lockdown) is
    %.2f. ',alpha/beta)

% lockdown
lockdown_period = 0; % period of lockdown in days
lockdown_infective_percentage = 0.30; % lockdown starts when this
    percentage of people is infected
lockdown_alpha = 0.04; lockdown_beta = 0.07; lockdown_gamma = 0.01; %
    parameters during the lockdown

% free day(s)
free_period = 14; % period of no restrictions
free_period_start = 20; % the day the free period starts
free_period_alpha = 0.3; free_period_beta = 0.125; free_period_gamma =
    0.01; % parameters during free days

% Calculation of the SIR values over time
[s,i,r,start_lockdown] = sir3(alpha,beta,gamma,s0,i0,r0,T,dt,
    lockdown_period,lockdown_infective_percentage,lockdown_alpha,
    lockdown_beta,lockdown_gamma,free_period,free_period_start,
    free_period_alpha,free_period_beta,free_period_gamma);

% Plot of the population division over time
t = 0:dt:T-dt; % time interval
```

```
23 plot(t,s,'b',t,i,'r',t,r,'g','LineWidth',2); grid on; xlabel('Days');
    ylabel('Fraction of individuals'); title('Population fractions in
    the basic SIR model over time'); legend('Fraction of susceptibles',
    'Fraction of infected','Fraction of recovered'); saveas(gcf,'
    FreeDaysSIR2.pdf');
```

In this code the function SIR3 is used, which is given below.

```
1  function [s,i,r,start_lockdown] = sir3(alpha,beta,gamma,s0,i0,r0,T,dt,
       lockdown_period,lockdown_infective_percentage,lockdown_alpha,
       lockdown_beta,lockdown_gamma,free_period,free_period_start,
       free_period_alpha,free_period_beta,free_period_gamma)
2      start_lockdown = 0;
3      s = zeros(1,T/dt); i=s; r=s; k1s = s; k1i = s; k1r = s; k2s = s;
           k2i = s; k2r = s; k3s = s; k3i = s; k3r = s; k4s = s; k4i = s;
           k4r = s;
4      s(1) = s0; i(1) = i0; r(1) = r0; lockdown = 0; % assume there has
           not been a lockdown yet
5      starting_alpha = alpha; starting_beta = beta; starting_gamma =
           gamma; starting_lockdown_period = lockdown_period;
6      for t = 1:(T/dt)-1
7          k1s(t) = (-alpha * s(t) * i(t) + gamma * r(t));
8          k1i(t) = (alpha * s(t) * i(t) - beta * i(t));
9          k1r(t) = (beta * i(t) - gamma * r(t));
10         k2s(t) = (-alpha * (s(t)+k1s(t)*dt/2) * (i(t)+k1i(t)*dt/2) +
               gamma * (r(t)+k1r(t)*dt/2));
11         k2i(t) = (alpha * (s(t)+k1s(t)*dt/2) * (i(t)+k1i(t)*dt/2) -
               beta * (i(t)+k1i(t)*dt/2));
12         k2r(t) = (beta * (i(t)+k1i(t)*dt/2) - gamma * (r(t)+k1r(t)*dt
               /2));
13         k3s(t) = (-alpha * (s(t)+k2s(t)*dt/2) * (i(t)+k2i(t)*dt/2) +
               gamma * (r(t)+k2r(t)*dt/2));
14         k3i(t) = (alpha * (s(t)+k2s(t)*dt/2) * (i(t)+k2i(t)*dt/2) -
               beta * (i(t)+k2i(t)*dt/2));
15         k3r(t) = (beta * (i(t)+k2i(t)*dt/2) - gamma * (r(t)+k2r(t)*dt
               /2));
16         k4s(t) = (-alpha * (s(t)+k3s(t)*dt) * (i(t)+k3i(t)*dt) + gamma
               * (r(t)+k3r(t)*dt));
17         k4i(t) = (alpha * (s(t)+k3s(t)*dt) * (i(t)+k3i(t)*dt) - beta *
               (i(t)+k3i(t)*dt));
18         k4r(t) = (beta * (i(t)+k3i(t)*dt) - gamma * (r(t)+k3r(t)*dt));
19         s(t+1) = (k1s(t)+2*k2s(t)+2*k3s(t)+k4s(t)) * dt/6 + s(t);
20         i(t+1) = (k1i(t)+2*k2i(t)+2*k3i(t)+k4i(t)) * dt/6 + i(t);
21         r(t+1) = (k1r(t)+2*k2r(t)+2*k3r(t)+k4r(t)) * dt/6 + r(t);
22         if (i(t+1)>lockdown_infective_percentage) && (lockdown==0) && (
               lockdown_period>0)
23             lockdown = 1;
24             alpha = lockdown_alpha; beta = lockdown_beta; gamma =
                   lockdown_gamma; lockdown_period = lockdown_period - dt;
                    start_lockdown = (t+1)*dt;
25             fprintf('Lockdown starting at day %.0f. ',start_lockdown)
26         elseif (lockdown_period>0) && (lockdown==1)
27             lockdown_period = lockdown_period - dt;
28         else
29             alpha = starting_alpha; beta = starting_beta; gamma =
                   starting_gamma; lockdown = 0;
```

```
30              lockdown_period = starting_lockdown_period; % comment for
                    only one lockdown
31          end
32          if (free_period/dt>0) && (t>free_period_start/dt) && (t<(
                free_period_start+free_period)/dt)
33              alpha = free_period_alpha; beta = free_period_beta; gamma =
                    free_period_gamma;
34          end
35      end
36  end
```

6.2.6 (Forward) Euler method on the Fatal SIR model

The Matlab code below yields a graph of the fatal SIR model by using (forward) Euler.

```
1  %% Fatal SIR model using (Forward) Euler
2  clear; close all; clc;
3
4  % Parameters
5  alpha = 0.1*1.728*10^(-6); % infection rate
6  beta = 1-0.00583565217; % recovery rate (= 1 - death rate for infected)
7  gamma = 0.001; % immunity loss rate
8  s0 = 1.728*10^7; i0 = 1; r0 = 0; T = 5000; % initial values and period
9  dt = 0.1; % length of the time interval (dt should hence divide T)
10 sbirth = 463; % number of new individuals per timestep capable of
       getting the virus (463 babies per day in the Netherlands)
11 sdeath = 0.00003389; % death rate for susceptibles (life expectancy:
       29.5k days)
12 ideath = 0.00583565217+0.00003389; % death rate for susceptibles (3.4%
       dies on average + life expectancy)
13 rdeath = 0.00003389; % death rate for recovered (life expectancy: 29.5k
        days)
14
15 % Calculation of the SIR values over time
16 [s,i,r] = sirBD(alpha,beta,gamma,s0,i0,r0,T,dt,sbirth,sdeath,ideath,
       rdeath);
17
18 % Total population
19 N = zeros(1,T/dt);
20 for t = 1:(T/dt)
21     N(t)=s(t)+i(t)+r(t);
22 end
23
24 % Plot of the population division over time
25 t = 0:dt:T-dt; % time interval
26 plot(t,s,'b',t,i,'r',t,r,'g',t,N,'y','LineWidth',2); grid on; xlabel('
       Days'); ylabel('Number of individuals'); title('Population in the
       fatal SIR model over time'); legend({'Number of susceptibles','
       Number of infected','Number of recovered','Total'}, 'Location', '
       east'); saveas(gcf,'FatalSIR2.pdf');
27 %{
28 % Computing fractions of the total population (note: this is not a
       closed system)
29 sfrac = s./(s0+i0+r0); ifrac = i./(s0+i0+r0); rfrac = r./(s0+i0+r0);
30 % Plot of the fraction of susceptibles, infected and recovered over
       time
```

```
t = 0:dt:T-dt; plot(t,sfrac,'b',t,ifrac,'r',t,rfrac,'g','LineWidth',2);
    grid on; xlabel('Days'); ylabel('Fraction of the total population');
    title('Fractions of the population in the SIR model with births and deaths over time'); legend('Fraction of susceptibles','Fraction of infected','Fraction recovered'); saveas(gcf,'FractionsSIR.pdf');
    % saving a pdf of the plot
%}
```

In this code the function sirBD is used, which is given below.

```
function [s,i,r] = sirBD(alpha,beta,gamma,s0,i0,r0,T,dt,sbirth,sdeath,ideath,rdeath)
    s = zeros(1,T/dt); i = s; r = s; s(1) = s0; i(1) = i0; r(1) = r0;
    for t = 1:(T/dt)-1 % for each timestep t determining s(t), i(t) and r(t)
        s(t+1) = s(t) + (-alpha*i(t)*s(t) + gamma*r(t) - sdeath*s(t) + sbirth) * dt;
        i(t+1) = i(t) + (alpha*i(t)*s(t) - beta*i(t) - ideath*i(t)) * dt;
        r(t+1) = r(t) + (beta*i(t) - gamma*r(t) - rdeath*r(t)) * dt;
    end
end
```

6.2.7 Comparing methods

The Matlab code below is used to compare the (forward) Euler, 4-stage Runge-Kutta and Dormand Prince method.

```
%% Comparing methods in the SIR model
clear; close all; clc;
format long % many more decimals

% Parameters
alpha = 0.3; beta = 0.045; mu = 0.2; s0 = 1-(10^(-2)); i0 = (10^(-2));
    r0 = 0.0; T = 200; dt = .01; % length of the time interval (dt
    should hence divide T)
fprintf('The value of the reproduction number R0 is %.2f. ',alpha/beta)
period = T; % for notation

% Forward Euler calculations
[s,i,r] = sircompare1(alpha,beta,mu,s0,i0,r0,T,dt);
Euler_end = i(end);
%{
% Forward Euler plot
t = 0:dt:T-dt; plot(t,s,'b',t,i,'r',t,r,'g','LineWidth',2); grid on;
    xlabel('Days'); ylabel('Fraction of individuals'); title('
    Population fractions in the basic SIR model over time'); legend('
    Fraction of susceptibles','Fraction of infected','Fraction of
    recovered'); saveas(gcf,'Euler_SIR_3.pdf');
%}

% Dormand Prince method calculations
t0 = 0; tmax = period; X0 = [s0; i0; r0]; options = odeset('RelTol',1e
    -8,'AbsTol',1e-8);
[ T,X ] = ode45( @(T,X) sircompare2(T,X,alpha,beta,mu), [t0 tmax], X0,
    options );
DP_end = X(end,2)
```

```
%{
% Dormand Prince method plot
plot(T,X(:,1),'b',T,X(:,2),'r',T,X(:,3),'g'); grid on; xlabel('Days'); ylabel('Fraction of individuals'); title('Population fractions in the basic SIR model over time (Runge–Kutta)'); legend('Fraction of susceptibles','Fraction of infected','Fraction of recovered'); % saveas(gcf,'PopulationstestSIR2.pdf');
%}

% Runge–Kutta 4/5 calculations
[u,v,w] = sircompare3(alpha,beta,mu,s0,i0,r0,period,dt); %u,v and w are s,i and r, respectively
RK_end = v(end);
%{
% Runge–Kutta 4/5 plot
t = 0:dt:period–dt; plot(t,u,'b',t,v,'r',t,w,'g','LineWidth',2); grid on; xlabel('Days'); ylabel('Fraction of individuals'); title('Population fractions in the basic SIR model over time'); legend('Fraction of susceptibles','Fraction of infected','Fraction of recovered'); saveas(gcf,'FreeDaysSIR1.pdf');
%}

%{
% Exact solution plot
lambda = alpha–mu+alpha*(s0+i0–1);
fprintf('The value of lambda is %.8f. ',lambda);
fplot(@(r) 1+(s0+i0–1)*(1–mu*r)–((lambda*i0)/(alpha*i0+(lambda–alpha*i0)*exp(–lambda*r))),[0 100],'b');
hold on
fplot(@(r) ((lambda)/(alpha+lambda*((lambda–alpha*i0)/(lambda*i0*exp((alpha*(s0+i0–1))/mu)))*exp(–lambda*r+((alpha*(s0+i0–1))/(mu))))),[0 100],'r')
hold off
%}

lambda = alpha–mu+alpha*(s0+i0–1);
Exact_end = ((lambda)/(alpha+lambda*((lambda–alpha*i0)/(lambda*i0*exp((alpha*(s0+i0–1))/mu)))*exp(–lambda*period+((alpha*(s0+i0–1))/(mu)))));
Euler_error = abs(Euler_end – Exact_end); DP_error = abs(DP_end – Exact_end); RK_error = abs(RK_end – Exact_end);
```

In this code the function SIRCOMPARE1 is used, which is given below.

```
function [s,i,r] = sircompare1(alpha,beta,mu,s0,i0,r0,T,dt)
    s = zeros(1,T/dt); i = s; r=s; s(1) = s0; i(1) = i0; r(1) = r0;
    for t = 1:(T/dt)
        s(t+1) = (–alpha * s(t) * i(t) – mu * s(t) + mu) * dt + s(t);
        i(t+1) = (alpha * s(t) * i(t) – mu * i(t))* dt + i(t);
        r(t+1) = (beta * i(t) – mu * r(t)) * dt + r(t);
    end
end
```

Moreover, in the code the function SIRCOMPARE2 is used, which is given below.

```
function dx2 = sircompare2(T,X,alpha,beta,mu)
    dx2 = zeros(3,1);
```

```
    dx2(1) = -alpha*X(1).*X(2) - mu*X(1) + mu;
    dx2(2) = alpha*X(1).*X(2) - mu*X(2);
    dx2(3) = beta*X(2) - mu*X(3);
end
```

Furthermore, in the code the function SIRCOMPARE3 is used, which is given below.

```
function [u,v,w] = sircompare3(alpha,beta,mu,s0,i0,r0,T,dt)
    s = zeros(1,T/dt); i=s; r=s; k1s = s; k1i = s; k1r = s; k2s = s;
        k2i = s; k2r = s; k3s = s; k3i = s; k3r = s; k4s = s; k4i = s;
        k4r = s;
    s(1) = s0; i(1) = i0; r(1) = r0;
    for t = 1:(T/dt)
        k1s(t) = (-alpha * s(t) * i(t) - mu * s(t) + mu);
        k1i(t) = (alpha * s(t) * i(t) - mu * i(t));
        k1r(t) = (beta * i(t) - mu * r(t));
        k2s(t) = (-alpha * (s(t)+k1s(t)*dt/2) * (i(t)+k1i(t)*dt/2) - mu
            * (s(t)+k1s(t)*dt/2) + mu);
        k2i(t) = (alpha * (s(t)+k1s(t)*dt/2) * (i(t)+k1i(t)*dt/2) - mu
            * (i(t)+k1i(t)*dt/2));
        k2r(t) = (beta * (i(t)+k1i(t)*dt/2) - mu * (r(t)+k1r(t)*dt/2));
        k3s(t) = (-alpha * (s(t)+k2s(t)*dt/2) * (i(t)+k2i(t)*dt/2) - mu
            * (s(t)+k2s(t)*dt/2) + mu);
        k3i(t) = (alpha * (s(t)+k2s(t)*dt/2) * (i(t)+k2i(t)*dt/2) - mu
            * (i(t)+k2i(t)*dt/2));
        k3r(t) = (beta * (i(t)+k2i(t)*dt/2) - mu * (r(t)+k2r(t)*dt/2));
        k4s(t) = (-alpha * (s(t)+k3s(t)*dt) * (i(t)+k3i(t)*dt) - mu * (
            s(t)+k3s(t)*dt) + mu);
        k4i(t) = (alpha * (s(t)+k3s(t)*dt) * (i(t)+k3i(t)*dt) - mu * (i
            (t)+k3i(t)*dt));
        k4r(t) = (beta * (i(t)+k3i(t)*dt) - mu * (r(t)+k3r(t)*dt));
        s(t + 1) = (k1s(t)+2*k2s(t)+2*k3s(t)+k4s(t)) * dt/6 + s(t);
        i(t + 1) = (k1i(t)+2*k2i(t)+2*k3i(t)+k4i(t)) * dt/6 + i(t);
        r(t + 1) = (k1r(t)+2*k2r(t)+2*k3r(t)+k4r(t)) * dt/6 + r(t);
    end
    u = s; v = i; w = r;
end
```

6.2.8 (Forward) Euler method on the SEIRP model.

The Matlab code below yields a graph of the compartment division in the SEIRP model by using the (forward) Euler method.

```
%% Euler method for the SEIRP model
clear; close all; clc; format long;

% Parameters
alpha_e = 0.4; alpha_i = 0.2; beta = 0.35; gamma = 0.01; kappa = 0.02;
    rho = 0.03; mu = 0.005836;
s0 = 1-(10^(-2)); e0 = 0.5*(10^(-2)); i0 = 0.5*(10^(-2)); r0 = 0.0; p0
    = 0.0; T = 10000; dt = 0.01;

% Calculation of the SIR values over time
[s,e,i,r,p] = seirp1(alpha_e,alpha_i,beta,gamma,kappa,rho,mu,s0,e0,i0,
    r0,p0,T,dt);

% Plot of the population division over time
```

```
t = 0:dt:T-dt; % time interval
plot(t,s,'b',t,e,'y',t,i,'r',t,r,'g',t,p,'m','LineWidth',2); grid on;
    xlabel('Days'); ylabel('Fraction of individuals');
title('Population fractions in the SEIRP model over time'); legend({'
    Fraction of susceptibles','Fraction of exposed','Fraction of
    infected','Fraction of recovered','Fraction of passed away'}, '
    Location', 'east');
saveas(gcf,'Euler_SEIRP_2.pdf'); % saving a pdf of the plot
```

In this code the function SEIRP1 is used, which is given below.

```
function [s,e,i,r,p] = seirp1(alpha_e,alpha_i,beta,gamma,kappa,rho,mu,
    s0,e0,i0,r0,p0,T,dt)
     s = zeros(1,T/dt); e=s; i=s; r=s; p=s; s(1) = s0; e(1) = e0; i(1) =
         i0; r(1) = r0; p(1) = p0;
     for t = 1:(T/dt)-1
         s(t+1) = (-alpha_e * s(t) * e(t) - alpha_i * s(t) * i(t) +
             gamma * r(t)) * dt + s(t);
         e(t+1) = (alpha_e * s(t) * e(t) + alpha_i * s(t) * i(t) - kappa
              * e(t) - rho * e(t)) * dt + e(t);
         i(t+1) = (kappa * e(t) - beta * i(t) - mu * i(t))* dt + i(t);
         r(t+1) = (beta * i(t) + rho * e(t) - gamma * r(t)) * dt + r(t);
         p(t+1) = (mu * i(t)) * dt + p(t);
     end
end
```

6.2.9 Prince Dormand method on the SEIRP model.

The Matlab code below yields a graph of the compartment division in the SEIRP model by using the Dormand Prince method. In Matlab this method has a built in function called ODE45. Furthermore, this code computes 2d and 3d phase plots and 2d and 3d direction fields.

```
%% Runge-Kutta method for the SEIRP model
clear all; clc;

%Parameters & Initial conditions
alpha_e = 0.5; alpha_i = 0.5; beta = 0.45; gamma = 0.01; kappa = 0.02;
    rho = 0.0; mu = 0.03;
S0 = 1-(10^(-2)); E0 = 0.5*(10^(-2)); I0 = 0.5*(10^(-2)); R0 = 0.0; P0
    = 0.0;
t0 = 0; tmax = 1000; X0 = [S0; E0; I0; R0; P0]; options = odeset('
    RelTol',1e-6,'AbsTol',1e-6);
[ T,X ] = ode45( @(T,X) seirp2(T,X,alpha_e,alpha_i,beta,gamma,kappa,rho
    ,mu), [t0 tmax], X0, options );
%{
plot(T,X(:,1),'b',T,X(:,2),'y',T,X(:,3),'r',T,X(:,4),'g',T,X(:,5),'m');
     grid on; xlabel('Days'); ylabel('Fraction of individuals');
title('Population fractions in the basic SIR model over time(Runge-
    Kutta)'); legend('Fraction of susceptibles','Fraction of infected',
    'Fraction of recovered');
%saveas(gcf,'PopulationstestSEIRP2.pdf'); % saving a pdf of the plot
%}
%A phase plot of S, E and I
plot3(X(:,1),X(:,2),X(:,3),'-o'); grid on; xlabel('s'); ylabel('e');
    zlabel('i'); saveas(gcf,'PhasePlotSEIRP.pdf'); % saving a pdf of
    the plot
```

In this code the function SEIRP2 is used, which is given below.

```
function dx2 = seirp2(T,X,alpha_e,alpha_i,beta,gamma,kappa,rho,mu)
    dx2 = zeros(5,1);
    dx2(1) = -alpha_e*X(1).*X(2)-alpha_i*X(1).*X(3) + gamma*X(4);
    dx2(2) = alpha_e*X(1).*X(2)+alpha_i*X(1).*X(3) - kappa*X(2)-rho*X
        (2);
    dx2(3) = kappa*X(2) - beta*X(3)-mu*X(3);
    dx2(4) = beta*X(3)+rho*X(2)-gamma*X(4);
    dx2(5) = mu*X(3);
end
```

6.2.10 4-stage Runge-Kutta method on the SEIRP model

The Matlab code below yields a graph of the compartment division in the SEIRP model by using the 4-stage Runge-Kutta method. Additionally, lockdowns and free days can be computed in this code by setting the variable(s) LOCKDOWN_PERIOD and FREE_PERIOD unequal to 0.

```
%% 4-stage RK method for the SEIRP model
clear; close all; clc; tic; % measure time

% Parameters
alpha_e = 0.4; alpha_i = 0.2; beta = 0.35; gamma = 0.01; kappa = 0.02;
    rho = 0.03; mu = 0.005836;
s0 = 1-(10^(-2)); e0 = 0.5*(10^(-2)); i0 = 0.5*(10^(-2)); r0 = 0.0; p0
    = 0.0; T = 100; dt = 0.01;

% lockdown
lockdown_period = 0; % period of lockdown in days
lockdown_infective_percentage = 0.01; % lockdown starts when this
    percentage of people is infected
ld_alpha_e = 0.04; ld_alpha_i = 0.04; ld_beta = 0.07; ld_gamma = 0.01;
    ld_kappa = 0.01; ld_rho = 0.01; ld_mu = 0.01; % values during
    lockdown

% free day(s)
free_period = 10; % period of no restrictions
free_period_start = 20; % the day the free period starts
fp_alpha_e = 0.6; fp_alpha_i = 0.6; fp_beta = 0.125; fp_gamma = 0.01;
    fp_kappa = 0.01; fp_rho = 0.01; fp_mu = 0.01;

% Calculation of the SIR values over time
[s,e,i,r,p,start_lockdown] = seirp3(alpha_e,alpha_i,beta,gamma,kappa,
    rho,mu,s0,e0,i0,r0,p0,T,dt,lockdown_period,
    lockdown_infective_percentage,ld_alpha_e,ld_alpha_i,ld_beta,
    ld_gamma,ld_kappa,ld_rho,ld_mu,free_period,free_period_start,
    fp_alpha_e,fp_alpha_i,fp_beta,fp_gamma,fp_kappa,fp_rho,fp_mu);

% Plot of the population division over time
t = 0:dt:T-dt; plot(t,s,'b',t,e,'y',t,i,'r',t,r,'g',t,p,'m','LineWidth'
    ,2); grid on; xlabel('Days'); ylabel('Fraction of individuals');
    title('Population fractions in the basic SIR model over time');
    legend('Fraction of susceptibles','Fraction of exposed','Fraction
    of infected','Fraction of recovered','Fraction of passed away');
    saveas(gcf,'LockdownSEIRP2.pdf'); % saving a pdf of the plot
```

In this code the function SEIRP3 is used, which is given below.

```
function [s,e,i,r,p,start_lockdown] = seirp3(alpha_e,alpha_i,beta,gamma
    ,kappa,rho,mu,s0,e0,i0,r0,p0,T,dt,lockdown_period,
    lockdown_infective_percentage,ld_alpha_e,ld_alpha_i,ld_beta,
    ld_gamma,ld_kappa,ld_rho,ld_mu,free_period,free_period_start,
    fp_alpha_e,fp_alpha_i,fp_beta,fp_gamma,fp_kappa,fp_rho,fp_mu)
    start_lockdown = 0;
    s = zeros(1,T/dt); e=s; i=s; r=s; p=s;
    k1s = s; k1e = s; k1i = s; k1r = s; k1p = s; k2s = s; k2e = s; k2i
        = s; k2r = s; k2p = s; k3s = s; k3e = s; k3i = s; k3r = s; k3p
        = s; k4s = s; k4e = s; k4i = s; k4r = s; k4p = s;
    s(1) = s0; e(1) = e0; i(1) = i0; r(1) = r0; p(1) = p0; lockdown =
        0; % assume there has not been a lockdown yet
    starting_alpha_e = alpha_e; starting_alpha_i = alpha_i;
        starting_beta = beta; starting_gamma = gamma; starting_kappa =
        kappa; starting_rho = rho; starting_mu = mu;
        starting_lockdown_period = lockdown_period;
    for t = 1:(T/dt)-1 % for each timestep t determining s(t), e(t), i(
        t), r(t) and p(t)
        k1s(t) = (-alpha_e * s(t) * e(t) - alpha_i * s(t) * i(t) +
            gamma * r(t));
        k1e(t) = (alpha_e * s(t) * e(t) + alpha_i * s(t) * i(t) -kappa
            * e(t) - rho * e(t));
        k1i(t) = (kappa * e(t) - beta * i(t) - mu * i(t));
        k1r(t) = (beta * i(t) + rho * e(t) - gamma * r(t));
        k1p(t) = (mu * i(t));
        k2s(t) = (-alpha_e * (s(t)+k1s(t)*dt/2) * (e(t)+k1e(t)*dt/2) -
            alpha_i * (s(t)+k1s(t)*dt/2) * (i(t)+k1i(t)*dt/2) + gamma *
             (r(t)+k1r(t)*dt/2));
        k2e(t) = (alpha_e * (s(t)+k1s(t)*dt/2) * (e(t)+k1e(t)*dt/2) +
            alpha_i * (s(t)+k1s(t)*dt/2) * (i(t)+k1i(t)*dt/2) -kappa *
            (e(t)+k1e(t)*dt/2) - rho * (e(t)+k1e(t)*dt/2));
        k2i(t) = (kappa * (e(t)+k1e(t)*dt/2) - beta * (i(t)+k1i(t)*dt
            /2) - mu * (i(t)+k1i(t)*dt/2));
        k2r(t) = (beta * (i(t)+k1i(t)*dt/2) + rho * (i(t)+k1i(t)*dt/2)
            - gamma * (r(t)+k1r(t)*dt/2));
        k2p(t) = (mu * (i(t)+k1i(t)*dt/2));
        k3s(t) = (-alpha_e * (s(t)+k2s(t)*dt/2) * (e(t)+k2e(t)*dt/2) -
            alpha_i * (s(t)+k2s(t)*dt/2) * (i(t)+k2i(t)*dt/2) + gamma *
             (r(t)+k2r(t)*dt/2));
        k3e(t) = (alpha_e * (s(t)+k2s(t)*dt/2) * (e(t)+k2e(t)*dt/2) +
            alpha_i * (s(t)+k2s(t)*dt/2) * (i(t)+k2i(t)*dt/2) -kappa *
            (e(t)+k2e(t)*dt/2) - rho * (e(t)+k2e(t)*dt/2));
        k3i(t) = (kappa * (e(t)+k2e(t)*dt/2) - beta * (i(t)+k2i(t)*dt
            /2) - mu * (i(t)+k2i(t)*dt/2));
        k3r(t) = (beta * (i(t)+k2i(t)*dt/2) + rho * (i(t)+k2i(t)*dt/2)
            - gamma * (r(t)+k2r(t)*dt/2));
        k3p(t) = (mu * (i(t)+k2i(t)*dt/2));
        k4s(t) = (-alpha_e * (s(t)+k3s(t)*dt) * (e(t)+k3e(t)*dt) -
            alpha_i * (s(t)+k3s(t)*dt) * (i(t)+k3i(t)*dt) + gamma * (r(
            t)+k3r(t)*dt));
        k4e(t) = (alpha_e * (s(t)+k3s(t)*dt) * (e(t)+k3e(t)*dt) +
            alpha_i * (s(t)+k3s(t)*dt) * (i(t)+k3i(t)*dt) -kappa * (e(t
            )+k3e(t)*dt) - rho * (e(t)+k3e(t)*dt));
        k4i(t) = (kappa * (e(t)+k3e(t)*dt) - beta * (i(t)+k3i(t)*dt) -
            mu * (i(t)+k3i(t)*dt));
        k4r(t) = (beta * (i(t)+k3i(t)*dt) + rho * (i(t)+k3i(t)*dt) -
```

```
            gamma * (r(t)+k3r(t)*dt));
        k4p(t) = (mu * (i(t)+k3i(t)*dt));
        s(t+1) = (k1s(t)+2*k2s(t)+2*k3s(t)+k4s(t)) * dt/6 + s(t);
        e(t+1) = (k1e(t)+2*k2e(t)+2*k3e(t)+k4e(t)) * dt/6 + e(t);
        i(t+1) = (k1i(t)+2*k2i(t)+2*k3i(t)+k4i(t)) * dt/6 + i(t);
        r(t+1) = (k1r(t)+2*k2r(t)+2*k3r(t)+k4r(t)) * dt/6 + r(t);
        p(t+1) = (k1p(t)+2*k2p(t)+2*k3p(t)+k4p(t)) * dt/6 + p(t);
        if (i(t+1)>lockdown_infective_percentage) && (lockdown==0) && (
            lockdown_period>0)
            lockdown = 1;
            alpha_e = ld_alpha_e; alpha_i = ld_alpha_i; beta = ld_beta;
                gamma = ld_gamma; kappa = ld_kappa; rho = ld_rho; mu =
                ld_mu;
            lockdown_period = lockdown_period - dt; start_lockdown = (t
                +1)*dt; fprintf('Lockdown starting at day %.0f. ',
                start_lockdown)
        elseif (lockdown_period>0) && (lockdown==1)
            lockdown_period = lockdown_period - dt;
        else
            alpha_e = starting_alpha_e; alpha_i = starting_alpha_i;
                beta = starting_beta; gamma = starting_gamma; kappa =
                starting_kappa; rho = starting_rho; mu = starting_mu;
                lockdown = 0;
            lockdown_period = starting_lockdown_period; % comment for
                only one lockdown
        end
        if (free_period/dt>0) && (t>free_period_start/dt) && (t<(
            free_period_start+free_period)/dt)
            alpha_e = fp_alpha_e; alpha_i = fp_alpha_i; beta = fp_beta;
                gamma = fp_gamma; kappa = fp_kappa; rho = fp_rho; mu =
                fp_mu;
        end
    end
end
```

6.2.11 (Forward) Euler method on the SEIRP model with vaccinations

The Matlab code below yields a graph of the compartment division in the SEIRP model with vaccinations by using the Euler method.

```
%% Euler method for the SEIRP model with vaccinations
clear; close all; clc; format long;

% Parameters
alpha_e = 0.4; alpha_i = 0.2; beta = 0.35; gamma = 0.01; kappa = 0.02;
    rho = 0.03; mu = 0.005836;
theta1 = 2*(10^(-8)); theta2 = 4*(10^(-10)); theta3 = 0.0; % theta =
    theta1*(t^2)*(dt^2) + theta2 * t * dt + theta3
s0 = 1-(10^(-2)); e0 = 0.5*(10^(-2)); i0 = 0.5*(10^(-2)); r0 = 0.0; p0
    = 0.0;
T = 200; dt = 0.01; % length of the time interval (dt should hence
    divide T)

% Calculation of the SIR values over time
[s,e,i,r,p] = seirpvac(alpha_e,alpha_i,beta,gamma,kappa,rho,mu,theta1,
    theta2,theta3,s0,e0,i0,r0,p0,T,dt);
```

```
12
13 % Plot of the population division over time
14 t = 0:dt:T-dt; plot(t,s,'b',t,e,'y',t,i,'r',t,r,'g',t,p,'m','LineWidth'
       ,2); grid on; xlabel('Days'); ylabel('Fraction of individuals');
15 title('Population fractions in the SEIRP model over time'); legend({'
       Fraction of susceptibles','Fraction of exposed','Fraction of
       infected','Fraction of recovered','Fraction of passed away'}, '
       Location', 'east'); saveas(gcf,'Euler_SEIRP_11.pdf'); % saving a
       pdf of the plot
```

In this code the function SEIRPVAC is used, which is given below.

```
1  function [s,e,i,r,p] = seirpvac(alpha_e,alpha_i,beta,gamma,kappa,rho,mu
       ,theta1,theta2,theta3,s0,e0,i0,r0,p0,T,dt)
2      s = zeros(1,T/dt); e = s; i = s; r = s; p =s; s(1) = s0; e(1) = e0;
           i(1) = i0; r(1) = r0; p(1) = p0;
3      for t = 1:(T/dt)-1 % for each timestep t determining s(t), i(t) and
           r(t)
4          theta = 1;
5          if (theta1*(t^2)*(dt^2) + theta2 * t * dt + theta3)<1
6              theta = (theta1*(t^2)*(dt^2) + theta2 * t * dt + theta3);
7          end
8          s(t+1) = (-alpha_e * s(t) * e(t) - alpha_i * s(t) * i(t) +
               gamma * r(t) -  theta * s(t)) * dt + s(t);
9          e(t+1) = (alpha_e * s(t) * e(t) + alpha_i * s(t) * i(t) - kappa
               * e(t) - rho * e(t)) * dt + e(t);
10         i(t+1) = (kappa * e(t) - beta * i(t) - mu * i(t))* dt + i(t);
11         r(t+1) = (beta * i(t) + rho * e(t) - gamma * r(t) + theta * s(t
               )) * dt + r(t);
12         p(t+1) = (mu * i(t)) * dt + p(t);
13     end
14 end
```