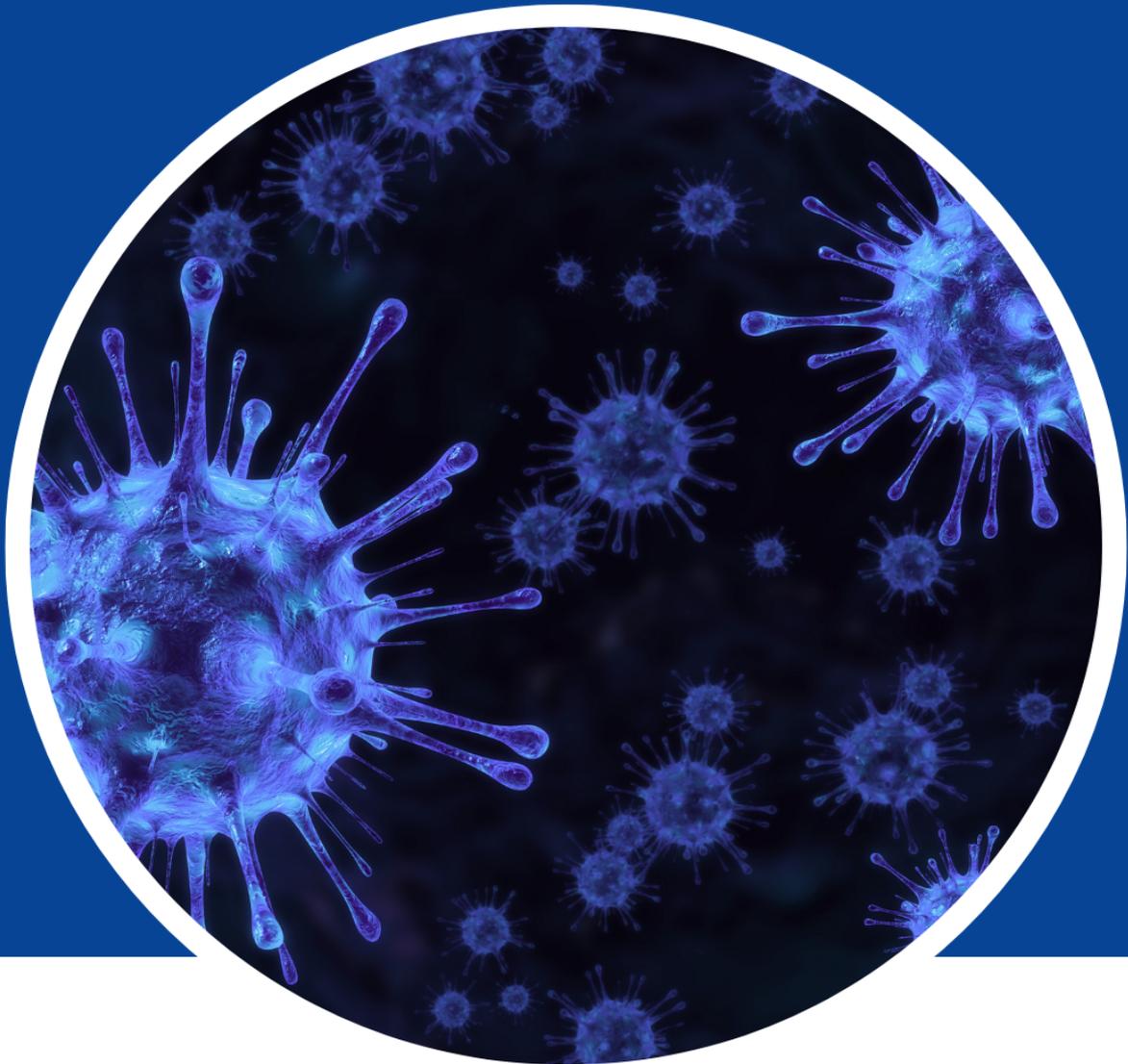


Matura Project by
Adriana Neumann

G4b

Mentored by
Dr. Thomas Preu



The SIR Model

Infectious Disease Modelling with
Differential Equations

Handed in December 12, 2022



■ **Kantonsschule Hottingen**

Table of Contents

PREAMBLE	5
ABSTRACT	ERROR! BOOKMARK NOT DEFINED.
1 INTRODUCTION	6
2 THE SIR MODEL	9
2.1 INFECTIOUS DISEASES	9
2.2 TYPES OF MODELS	10
2.3 THE BASIC SIR MODEL	11
2.4 THE SIR MODEL WITH DEMOGRAPHY	12
2.5 ASSUMPTIONS OF THE SIR MODEL	12
3 THE BASIC REPRODUCTION NUMBER R_0	13
3.1 MEANING	13
3.2 THE EQUILIBRIUM STATES	14
3.3 STABILITY PROPERTIES	16
4 EXTENSION OF THE BASIC SIR MODEL	16
4.1 THE SI MODEL – LIFELONG INFECTION	16
4.2 THE SEIR MODEL – INTRODUCING A CARRIER STATE	17
5 MATHEMATICAL METHODS	19
5.1 RUNGE-KUTTA METHODS	19
5.2 FEHLBERG METHOD	20
5.3 BOGACKI-SHAPMINE METHOD	21
5.4 ERRORS AND RUNGE KUTTA METHODS	21
5.5 EFFICIENCY OF NUMERICAL INTEGRATION METHODS	23

6	COMBINING MATHEMATICS AND BIOLOGY	24
6.1	METHOD	24
7	RESULTS	26
7.1	CHANGING PARAMETER VALUES	26
7.2	ERROR ANALYSIS AND ROUNDING	27
7.3	EFFICIENCY NUMBERS	30
8	DISCUSSION	33
8.1	THE EFFECT OF CHANGING PARAMETRIZATION	33
8.2	ERROR ANALYSIS AND THE EFFECT OF ROUNDING	34
8.3	ANALYSIS OF EFFICIENCY NUMBERS	35
9	CONCLUSION	37
10	REFLECTION	38
11	BIBLIOGRAPHY	39
12	TABLE OF FIGURES	42
13	LIST OF TABLES	42
14	DECLARATION	43
15	WORK JOURNAL	44
	APPENDIX I	47
	R-CODE	47
	DATA PARAMETER EXPERIMENTS	51

APPENDIX II _____ **67**

EXCEL SOLUTIONS AND ERROR DATA _____ **67**

DATA ROUNDING EXPERIMENTS _____ **81**

Preamble

Before I chose a specific topic to base my project on, I knew I wanted it to relate to mathematics. The way in which mathematics can be applied to real life problems by employing mathematical models has always fascinated me. These models come in a variety of shapes and sizes and allow us to represent natural phenomena using variables and equations. Finding patterns in seemingly chaotic behaviour is one of mathematics' many virtues.

Towards the end of 2021, my math teacher, Mr. Preu, introduced my class to differential equations. Since there exists a method of disease modelling based on differential equations, namely the SIR model, he thought it might be worth looking into as a potential topic. The SIR model divides the population into susceptible, infected, and recovered and uses 3 differential equations to calculate the changes in size of said three groups short term.

During the Coronavirus pandemic, disease modelling and the predictions made using it were topics widely talked about on the news and therefore on my radar already.

As soon as I started delving into preliminary research regarding the SIR model, I knew it was something I'd be willing to research more intensively, my final project being the perfect opportunity to do so. I was intrigued by the prospect of executing my own simulations and parsing through the generated data. I didn't want my project to only focus on the disease model from a mathematical perspective, but also sought to interpret it from an epidemiological standpoint.

Firstly, I want to thank my math teacher Mr. Preu for suggesting the topic in the first place. I greatly appreciate the effort he invested in being my mentor and his readiness to answer any questions I had. I'd also like to thank Professor Kuyos at the University of Zurich for giving me useful tips for disease simulating and helpful insights.

1 Introduction

Infectious diseases have accompanied the human species throughout recorded history. Their emergence and re-emergence constitute a worldwide health challenge. Some of these diseases, such as Polio, can inflict mortality on their hosts. HIV, for example, has accumulated a death toll of over 40 million people since its widespread emergence in 1981 (WHO, 2022). More than 90% of deaths resulting from infectious diseases can be attributed to a relatively small number of diseases (Ashcroft, Bonhoeffer, Kouyos, Regoes and Stadler, 2020, p.12).



Figure 1: Portrait of Daniel Bernoulli in 1750 (Britannica T. E., 1998)

To improve control over and ultimately eradicate an infection from a population, one can analyze infectious diseases and develop strategies to mitigate their harm. Some of these strategies include vaccination and quarantine. Mathematical models can be used as a means of researching diseases and are employed in epidemiology and many other fields. They can predict the population-level dynamics of an illness or the impact of vaccination on the spread of infection. The models discussed in this paper are based on systems of differential equations that can be solved using a method called numerical integration, which means employing derivatives of initial values (initial conditions) to extrapolate subsequent values. Numerical integration encompasses a broad range of algorithms, a few of which will be discussed in later sections.

The first mathematical model designed to investigate an infectious disease was developed by the Swiss Mathematician Daniel Bernoulli in 1766 (figure 1). Born in Groningen, he studied medicine in Heidelberg, Strasbourg, and Basel. His paper on differential equations and the physics of flowing water granted him a position at the influential Academy of Sciences in St. Petersburg, Russia (Britannica, 1998). In the context of Infectious diseases, he is known for studying the impact of variolation¹ on smallpox mortality in France. Bernoulli addressed the impact of variolation on the French population using a modeling study based on differential equations (Ashcroft et al., 2020, p.18).

¹ Variolation is an obsolete method of immunization patients against smallpox by infecting them with substance from the pustules of patients with a mild form of the disease (Britannica, 1998).

Another individual who made valuable contributions to the study of infectious disease dynamics was Ronald Ross (figure 2, left). He studied medicine, wrote novels, poetry and plays, and composed music. Being born in India, he became an officer in the Indian Medical Service. (Britannica, 1998). Concerning infectiology, he independently discovered the mass action term² and extended this insight into his *theory of happenings* in which he explicitly defines the dynamical feedback of infectious diseases (Ashcroft et al., S.25). Ross' mathematical contributions laid the framework for Kermack and McKendrick's paper, in which they first introduced the model this paper seeks to analyze.



Figure 2: Ronald Ross, Anderson G. McKendrick, and William O. Kermack (Britannica T. E., 1998), (Bacaer N. , 2011)

In a very famous paper published in 1927, William O. Kermack, who was a superintendent of the Laboratory of the Royal College of Physicians of Edinburgh, and Anderson G. McKendrick, an employee of Kermack's, extended Ross' mathematical framework (Ashcroft et al., 2020, p.25). They introduced what is today known as the SIR model. The SIR model and its variants are one of the most important instruments used in theoretical epidemiology.

This begs the question; **how do the elements of the SIR model affect predictions made about disease dynamics, from both a mathematical and epidemiological perspective?**

The objective of this paper is to give insight into the basics of compartmental disease modeling³ through theory and implementation using numerical integration methods pertaining to the

² The mass action term signifies that the number of new infections depends on the number of infected and susceptibles and thus formalizes the feedback of an infection on itself (Bonhoeffer, Regoes, Stadler, Kouyos, Ashcroft, 2020, p.21).

³ Disease modeling in which the population is separated into compartments such as susceptible, infected, and recovered

Runge-Kutta family of integrators. In this work, numerical experiments to find out how the model parameters β and γ affect the S, I and R curves, how epidemiological concepts such as the basic reproduction number influence the dynamics and how elements such as time stepping, rounding and computational cost of Runge-Kutta integrators interact when solving the basic SIR model are conducted. How a change in step size or rounding influences accuracy is analyzed and comparisons of the efficiency of Runge-Kutta methods are made. I hypothesize that an increase in the basic reproduction number R_0 will lead to a higher number of infected at the disease's peak. In terms of error analysis, I hypothesize that smaller step sizes lead to lower errors per step. I also predict that methods that use the least total amount of function evaluations to approximate a solution, compute the most efficiently.

Chapters in this paper focus solely on explicit Runge-Kutta methods and experiments are restricted to solving the basic SIR model without demography. The tools used to simulate the SIR models are the ODE package and its Runge-Kutta solvers in R and manual implementation in Excel. Parameter values are either chosen by the author or derived from real data.

The theoretical part of this paper formulates the model equations governing the SIR model and its variants and introduces basic concepts such as the basic reproduction number and endemic equilibrium. It also explains the relevance of numerical integration methods and the way they work. After this follows a section in which the method used in the numerical experiments is presented and the results are shown, discussed, and compared with theory. Lastly, further research topics extending the contents of this paper are presented, followed by a reflection.

2 The SIR Model

2.1 Infectious diseases

The term “disease” includes a variety of ailments ranging from cancer to the flu. Overall, epidemiologists classify diseases according to several factors. They can either be infectious or noninfectious (Keeling & Rohani, 2007, p. 2). In his “*theory of happenings*”, Ronald Ross (1917, S.507) explicitly defined what sets infectious diseases apart from other illnesses; “different kinds of happenings may be separated into two classes, namely a) those in which the frequency of the happening is independent of the number of individuals already affected; and b) those in which the frequency of the happening depends on this quantity. To class b) belong infectious diseases.” This describes the dynamical feedback that constitutes a fundamental characteristic of infectious diseases.

Infectious diseases can be categorized according to their sources. Infecting pathogens can either be micro- or microparasitic. The former entails small (usually single-cell organisms), whereas the latter are larger. Microparasites include viruses and bacteria while their counterpart partially consists of parasitic worms such as helminths or flukes (Keeling & Rohani, 2007, S. 1).



Figure 3: Helminths and flukes (Hoffmann, Brindley, & Berriman, 2014), (Britannica, 1998)

Globally, there are about 1'415 known pathogens that cause diseases in humans, 53% of which are microparasites (Cleaveland, Laurenson, & Taylor, 2001).

Within the family of micro- or microparasitic infection, one can subdivide based on whether transmission happens directly or indirectly (Keeling & Rohani, 2007, p. 4). Direct transmission occurs through close contact with an infected person. Most microparasitic diseases behave in such a way. Examples include influenza, HIV, and measles, cholera being an exception. Indirectly transmitted sicknesses are passed via the environment (eg. water). The disease

models discussed in subsequent sections are geared towards the inspection of directly transmitted, microparasitic infectious diseases.

2.2 Types of models

Infectious disease models used in epidemiology can generally be divided into two classes; *deterministic* and *stochastic* (Dadlani, et al., 2013, p. 1). In deterministic models, the population is separated into smaller groups or compartments, each one of these groups illustrating a specific stage in the epidemic (Dadlani, et al., 2013, p. 2). Models like these can be expressed mathematically using systems of differential equations that try to represent what happens on average when a pathogen is introduced to the population. In mathematics, the word deterministic denotes the fact that no randomness occurs during the development of future states. (Darling, 2016). This means that a deterministic system uses earlier states of the system to determine later states. If something is strongly deterministic, a certain initial condition will inevitably lead to a certain outcome. Sometimes, the system can be tamely chaotic and small deviations or errors in data can lead to large deviations in predictions. Stochastic systems or a stochastic process entails randomness. More specifically, future states cannot be predicted from past ones. The flipping of a coin would be an example of a stochastic system.

The SIR model falls into the category of deterministic epidemic models and best describes a situation in which a pathogen causes illness for a short period (days or weeks) before the host attains lifelong immunity (Keeling & Rohani, 2007, p. 16). SIR stands for **S**usceptible (unexposed to illness), **I**nfected (colonized by pathogen and infectious) and **R**ecovered (pathogen has cleared host). This denotes the compartments the model divides the total population into.

2.3 The Basic SIR Model

If we regard the simplest possible case, meaning we neglect birth and death rates, we only take transitions from $S \rightarrow I$ and $I \rightarrow R$ into consideration, as seen in figure 4. This means we must find a way to mathematically express the movement of individuals from one group to another, which can be done using differential equations.



Figure 4: The movement of individuals from one host group to the other (UCLA, 2020).

An infected individual can only move to the recovered class after it has successfully fought the infection and therefore isn't infectious anymore. (Keeling & Rohani, 2007, p. 16). Through clinical data, scientist find an average **infectious period** p , e.g., the amount of time spent in the infected class and can derive the **recovery rate** γ , which is defined as $1/p$ and is seen as constant for simplicity purposes. The transition from **S** to **I** involves disease transmission, which depends on a) the amount of infected b) the contact rate between individuals and c) the probability of transmission given contact between and infected and susceptible. The transmission rate β represents the average amount of secondary infections caused by an infected and is defined as the per capita contact rate times the probability of infection (Keeling & Rohani, 2007, p. 18).

Let's define k as the contact rate per individual and c as the probability of infection given contact between infected and susceptible and N as the total population wherein there are Y infected, X susceptible and R recovered individuals. If we multiply the contact rate k with Y/N , we get the average amount of infected a susceptible meets in a given time. Multiplying $k \times Y/N$ with c yields the rate at which one susceptible gets infected or $\beta \times Y/N$. This term is also known as the force of infection λ (Keeling & Rohani, 2007, S. 18). The product of the force of infection and X yield the total rate of transmission to the entire susceptible population.

Using the information gathered above we can derive differential equations 1.1 – 1.3, one for each compartment (Keeling & Rohani, 2007, p. 18);

$$\frac{dX}{dt} = -\beta \frac{Y}{N} X, \quad (1.1)$$

$$\frac{dY}{dt} = \beta \frac{Y}{N} X - \gamma Y, \quad (1.2)$$

$$\frac{dR}{dt} = Y\gamma. \quad (1.3)$$

It is noteworthy that the transmission term $\beta \times Y/N \times X$ occurs in both differential equations 1.1 and 1.2. This is because susceptible individuals directly move from X to Y in our model. Similarly, the term $\gamma \times Y$ appears in both 1.2 and 1.3.

For scaled variables (fraction of total population), eq. 2.1-2.3 hold (Keeling & Rohani, 2007, p. 19);

$$\frac{dS}{dt} = -\beta SI, \quad (2.1)$$

$$\frac{dI}{dt} = \beta SI - \gamma I, \quad (2.2)$$

$$\frac{dR}{dt} = \gamma I. \quad (2.3)$$

In the following sections we will mostly use scaled variables with constant N, thus $S+I+R = 1$.

2.4 The SIR Model with demography

Let's look at what happens once we introduce demography e.g., death and birth rates. For this we introduce the parameters μ = birth rate/per capita death rates for S, I and R. This yields equations 3.1-3.3 (Keeling & Rohani, 2007, p. 27);

$$\frac{dS}{dt} = \mu - \beta SI - \mu S, \quad (3.1)$$

$$\frac{dI}{dt} = \beta IS - \gamma I - \mu I, \quad (3.2)$$

$$\frac{dR}{dt} = \gamma I - \mu R. \quad (3.3)$$

If we add all three equations together, we notice that $dS/dt + dI/dt + dR/dt = 0$, hence the total population size stays the same. The expected life expectancy per individual is $1/\mu$. (Bjornstad, 2018, p. 3)

2.5 Assumptions of the SIR model

The assumptions of eq. 1.1-3.3 are as follows (Ashcroft et al., 2020, S.30-31);

- Constant parameters; all parameters of the model are assumed to be constant. In reality, β can depend on the season.
- Mass action kinetics; the overall transmission is proportional to the size of susceptible and infected, that is SI. This is justified if both host types are well mixed and homogenous.

- Infections cause acute morbidity, not mortality. This is reasonable for diseases like chickenpox but not for others like Ebola.
- Infected individuals move directly into the infectious class (unlike the SEIR model; see later subsection).
- The model assumes that recovered individuals have lifelong immunity (unlike SI model; see later subsection).

3 The Basic Reproduction Number R_0

3.1 Meaning

How does one determine if an epidemic will occur or if the disease will fail to spread? To answer this question, we start by looking at dl/dt at the very beginning, denoting our initial I and S as I_0 and S_0 . Initially, if $dl/dt > 0$, the infection grows.

We write;

$$\beta S_0 I_0 - \gamma I_0 > 0 \Rightarrow \beta S_0 > \gamma \Rightarrow \frac{\beta S_0}{\gamma} > 1 \quad (4)$$

In epidemiology, β/γ denotes the Basic Reproduction Number R_0 (Keeling & Rohani, 2007, p. 19-20) and is understood as;

the average number of secondary cases arising from an average primary case in an entirely susceptible population ($S_0=1$).

It essentially measures the maximum reproductive potential for an infection.

From inequality 4, we can use $R_0 = \beta/\gamma$ to express the condition on disease invasion;

if $R_0 > 1$, disease invasion occurs

if $R_0 < 1$, disease invasion does not occur

This also means that if S_0 remains under γ/β , widespread infection cannot take place. This can be achieved by immunization, for example through vaccination campaigns. We can interpret this result as requiring the removal rate γ/β , or the inverse of R_0 , to be small enough for the disease to spread (Keeling & Rohani, 2007, p. 20). For the basic SIR model with demography, $R_0 = \beta/(\gamma + \mu)$. This value is generally similar to, but always smaller than, R_0 for a closed population. The natural mortality rate actively reduces the average time an individual spends in the infectious class; instead of $1/\gamma$, it's $1/(\gamma + \mu)$ (Keeling & Rohani, 2007, p. 28)

The parameters that govern the ordinary differential equations⁴ for S, I and R also control R_0 . This means that by decreasing the infectious period $1/\gamma$ or $1/(\gamma+\mu)$, for example through medicine or other types of medical intervention or decreasing β , R_0 is reduced. Measures that result in a smaller β include decreasing the contact rate of the host population (for example through quarantine) or decreasing the probability of transmission during contact (for example through mask-wearing). Mathematically, this means that the rate at which new cases are initially created, reduces as well. Visually, this results in the flattening of the Infectious curve – it isn't as steep, as seen in figure 5.

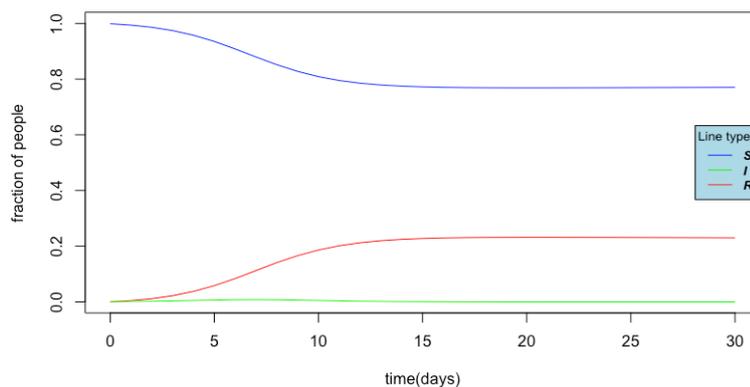


Figure 5: Numerical simulation of an Influenza outbreak in R, over 30 days. (based on Mahaffy, 2018). The initial conditions are $S=0.999$, $I=0.001$ and $R=0$. Because R_0 is just above 1, the Infectious curve is very flat with a maximum of 0.008 infected.

A host population is rarely fully susceptible, e.g., some individuals will be immune due to prior infection or because of vaccination. This is where the **effective reproduction number $R(E)$** comes into play. $R(E)$ denotes the average number of secondary cases per infectious individual in a non-fully susceptible population and can be approximated by multiplying the basic reproduction number R_0 with the fraction of susceptible in the host population; $R(E) = R_0 S$ (Baratt, Kirwan, & Shantikumar, 2009). If $R_0 > 1$, as soon as $R(E)=1$, the disease becomes endemic.

3.2 The Equilibrium states

If we want to explore what will eventually happen to a population after a pathogen is introduced, we figure out when the system is at equilibrium, e.g., $dS/dt = dI/dt = dR/dt = 0$.

⁴ Ordinary differential equations or ODEs are equations containing a function and its derivatives. It is usually written in the form of $y'=f(x,y)$ or $dy/dx(x)=f(x(y(x)))$ (Wai, Choon, & Idris, 2021, p. 26).

It is easy to discern the disease-free equilibrium. In this scenario, the disease has gone extinct, and everyone is susceptible, namely $(S_E, I_E, R_E) = (1, 0, 0)$.⁵ (Keeling & Rohani, 2007, p. 28). Establishing the endemic equilibrium, which occurs after the pathogen has run its course, requires more effort. We start by setting the equation from system 3 for the infectives to zero after factoring for I;

$$I(\beta \times S - (\gamma + \mu)) = 0 \quad (5)$$

This is satisfied whenever $I=0$ or $S = (\gamma + \mu) / \beta$. The first condition is the disease-free equilibrium and the second is the endemic equilibrium. If we solve eq. 5 for I using $S = (\gamma + \mu) / \beta$ in eq. 3.1, we get $I = \mu / (\gamma + \mu) - \mu / \beta$. Now utilizing $S_E + I_E + R_E = 1$, we can get an expression for R.

The endemic equilibrium is therefore given by eq. 6.1-6.3;

$$S_E = \frac{\gamma + \mu}{\beta}, \quad (6.1)$$

$$I_E = \frac{\mu}{\gamma + \mu} - \frac{\mu}{\beta}, \quad (6.2)$$

$$R_E = 1 - \frac{\gamma + \mu}{\beta} - \frac{\mu}{\gamma + \mu} + \frac{\mu}{\beta}. \quad (6.3)$$

Using $R_0 = \beta / (\gamma + \mu)$, we can simplify, which yields eq. 7.1-7.3;

$$S_E = \frac{\gamma + \mu}{\beta}, \quad (7.1)$$

$$I_E = \frac{\mu}{\gamma + \mu} - \frac{\mu}{\beta}, \quad (7.2)$$

$$R_E = 1 - \frac{\gamma + \mu}{\beta} - \frac{\mu}{\gamma + \mu} + \frac{\mu}{\beta}. \quad (7.3)$$

The endemic equilibrium is biologically only feasible if $R_0 > 1$ due to the universal condition on the non-negativity of S, I and R and shows that in the SIR model with demography, if an epidemic sweeps through a population, not all individuals become infected (Keeling & Rohani, 2007, p. 29). This does not imply however, that those who weren't infected were immune. Instead, this happens because new susceptible individuals are constantly being born into the population (Ashcroft et al., 2020, p.32).

⁵ S_E, I_E, R_E denote the S, I and R values at equilibrium.

3.3 Stability Properties

Now that we know the conditions surrounding the equilibrium states, we want to find out the likeliness of observing them. One can use mathematical stability analysis for each equilibrium point (Keeling & Rohani, 2007, p. 30). When looking at the rates of change of the variables when each variable is slightly shifted away from its equilibrium point, one can discern its stability; if an equilibrium state is stable, it can withstand small perturbations, e.g., the rate of change doesn't "blow up".

For our two equilibrium states, this yields (Keeling & Rohani, 2007, p. 29);

The endemic equilibrium is stable only if $R_0 > 1$

The disease-free equilibrium is stable only if $R_0 < 1$

This makes sense since a disease can only grow in the long term if each infected host passes on the infection to more than one individual. If fewer secondary infections take place, the pathogen doesn't spread.

4 Extension of the basic SIR Model

4.1 The SI Model – Lifelong infection

The SI model distinguishes between two classes: susceptible and infected. Just like in the SIR model, susceptible individuals become infected through disease transmission and move to the infectious group. Both infected and susceptible hosts are lost through death. In the SI model, a disease leaves the individual infected for the rest of their lives, this means they never recover. This holds true for diseases such as herpes. We still assume $S+I=1$ and that the total population size remains constant.

The SI equations are presented in eq. 8.1 and 8.2;

$$\frac{dS}{dt} = \mu - \mu S - \beta SI, \quad (8.1)$$

$$\frac{dI}{dt} = \beta SI - \mu I. \quad (8.2)$$

Deriving the basic reproduction number for the SI model yields $R_0 = \beta/\mu$, since the infectious period becomes $1/\mu$, or life expectancy.

The disease-free equilibrium is given by $(S_E, I_E) = (1, 0)$. The endemic equilibrium by eq. 9.1-9.2;

$$S_E = \frac{1}{R_0}, \quad (9.1)$$

$$I_E = 1 - \frac{1}{R_0}. \quad (9.2)$$

Just like in the SIR model, not all susceptible individuals become infected at equilibrium. The same stability properties from above hold.

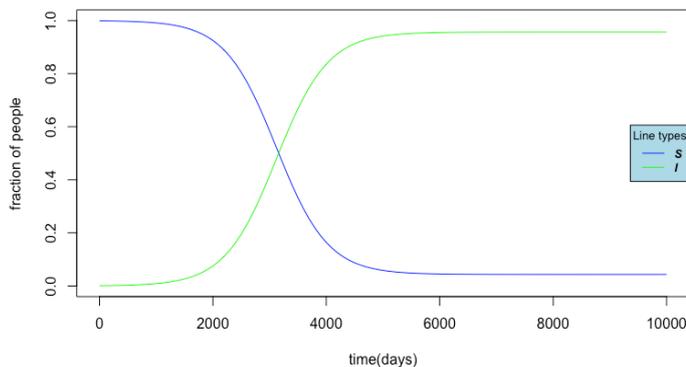


Figure 6: Numerical simulation of SI model herpes outbreak in R, over 10'000 days for initial $S=0.999$, $I=0.001$, and $R=0$ and $\beta=0.0285$ (based on Amalia S. Magaret, 2016)

4.2 The SEIR Model – Introducing a carrier state

The SEIR refines the SIR model by taking a latency period into consideration. After a susceptible individual is infected, a period ensues during which the pathogen reproduces rapidly. At this stage, the individual is *infected*, but not yet *infectious* (Keeling & Rohani, 2007, p. 41). We introduce a new host class, E , which illustrates this *infected but not infectious* state, and the latency period $1/\sigma$. We still assume $S+E+I+R=1$ and that the total population size remains constant. The SEIR equations are presented in eq. 10.1-10.4;

$$\frac{dS}{dt} = \mu - (\beta I + \mu)S, \quad (10.1)$$

$$\frac{dE}{dt} = \beta SI - (\mu + \sigma)E, \quad (10.2)$$

$$\frac{dI}{dt} = \sigma E - (\mu + \gamma)I, \quad (10.3)$$

$$\frac{dR}{dt} = \gamma I - \mu R. \quad (10.4)$$

The expression for R_0 changes and becomes more complicated due to the death of some individuals in the carrier class E . These individuals do not contribute to the growth of the infected group. However, this effect is often negligible since the latency period is oftentimes far smaller than life expectancy, so in the case where $\sigma \rightarrow \infty$ we can say that $R_0 = \beta / (\gamma + \mu)$.

The disease-free equilibrium is given by $(S_E, E_E, I_E, R_E) = (1, 0, 0, 0)$. The endemic equilibrium is given by eq. 11.1-11.4;

$$S_E = \frac{\mu + \sigma}{R_0 \sigma}, \quad (11.1)$$

$$E_E = \frac{\mu}{\mu + \sigma} - \frac{\mu}{R_0 \sigma}, \quad (11.2)$$

$$I_E = \frac{\mu}{\beta} \left(\frac{R_0 \sigma}{\mu + \sigma} - 1 \right), \quad (11.3)$$

$$R_E = 1 - S_E - E_E - I_E. \quad (11.4)$$

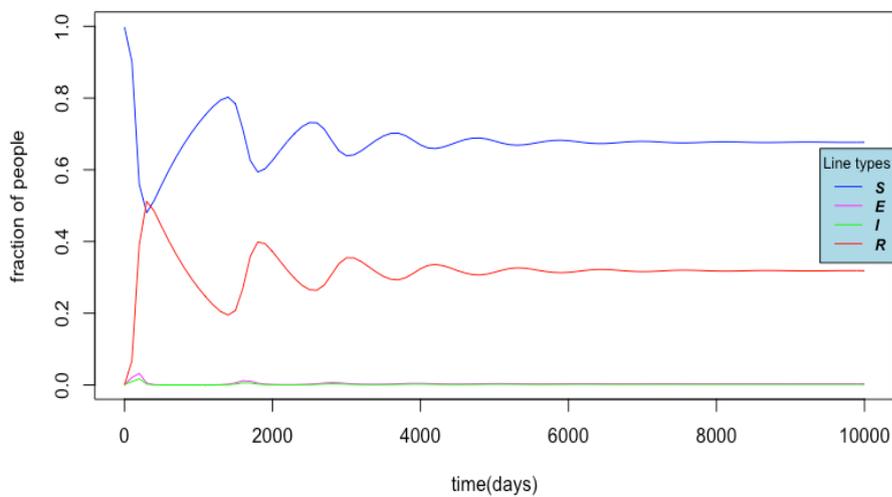


Figure 7: Numerical Simulation of SEIR model in R. The initial values are $S=0.998$, $E=0.001$, $I=0.001$ and $R=0$. The parameters $\beta=0.3$, $\gamma=0.2$, $\mu=0.001$ and $\sigma=0.1$ were chosen by the author. The graph shows oscillations due to the time delay between infection and infectiousness.

5 Mathematical Methods

5.1 Runge-Kutta Methods

This chapter focuses on methods that derive a solution to the SIR equations outlined in the previous chapters. Because of the complexities that arise in the 3 differential equations, it is rather difficult to obtain explicit functions that predict the evolution of S, I and R using analytical methods. Instead, one can use numerical integration (Keeling & Rohani, 2007, p. 25). Numerical integration entails using an algorithm to approximate future values using initial states and their slopes, stepwise. The Euler method is the easiest algorithm amongst a family of numerical integrators, the so-called Runge-Kutta family of integrators, because it only uses one function evaluation per step. Runge-Kutta methods are all “one-step methods”, which means the approximation for the next point $t + \Delta t$ is obtained by using information (slopes) only from the previous point. Some RK methods use function-evaluation information at points between t and $t + \Delta t$ but discard this information after the approximation at $t + \Delta t$ is derived (Kong, Siau, & Bayen, 2020). Standard software packages such as R offer solvers based on Runge-Kutta algorithms to solve ordinary differential equations.

A Runge-Kutta method can be expressed generally as in eq. 12.:

$$y_{n+1} = y_n + h \sum_{i=1}^s \gamma_i k_i \quad (12)$$

where y_n is the initial value, h the step size, γ_i coefficients depending on the specific RK method and k_i the slopes (Bogacki & Shampine, 1989).

Some ODE solvers have embedded error estimation methods, which are used to adapt the step size according to error tolerances and make the computation more efficient. The objective of an adaptive step size method is to stay within some error constraint with minimum computational effort (Wai, Choon, & Idris, 2021, p. 26). There are two RK adaptive step methods the next sections see to analyze further; the Runge-Kutta-Fehlberg method (ode45 solver) and Bogacki- Shampine method (ode23 solver).

5.2 Fehlberg Method

The **Fehlberg method** includes a fourth order and fifth order formula and was developed by Erwin Fehlberg in a 1969 NASA report (Fehlberg, 1969). It is implemented in R as the ode45 solver. The fourth order solution is used to further the solution, whereas the fifth order solution is used for local error estimation. The Fehlberg formulas are defined in eq. 13 and 14;

$$k_i = h_n f \left(y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j, t_n + \alpha_i h_n \right), i = 1, \dots, 6 \quad (13)$$

$$y_{n+1} = y_n + \sum_{i=1}^6 \gamma_i k_i \quad (14)$$

Where α_i and β_i are coefficients given in figure 8, and h_n is the time step (Zhang, 2015).

α_i	β_{ij}					γ_i	γ_i^*
0						$\frac{16}{135}$	$\frac{25}{216}$
$\frac{1}{4}$	$\frac{1}{4}$					0	0
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$				$\frac{6656}{12825}$	$\frac{1408}{2565}$
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$			$\frac{28561}{56430}$	$\frac{1408}{2565}$
1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$		$-\frac{9}{50}$	$-\frac{1}{5}$
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	$\frac{2}{55}$	0

Figure 8: Coefficients of Fehlberg's formulas, γ_i and γ_i^* are the weighted coefficients of the fifth and fourth order formulas (Hull, Enright, Fellen, & Sedgwick, 1972).

α_i indicates the intermediate step sizes that are taken; for both the fifth and fourth order method, $\frac{1}{4}$ of a full step is taken initially using the slope (k_1) at the initial value y_n . At this point, $\frac{3}{8}$ of a full step is taken using the slope at $t_n + \frac{1}{4} h_n$. The taking of intermediate steps is iterated 6 times in total, until all slopes k_1-k_6 are obtained. At the end, a linear combination of these 6 slopes is used to derive both a fifth and fourth order solution using weighted coefficients (last two columns of table 1). The estimated local error is estimated using the difference of the fifth and fourth order formulas.

Because Fehlberg's method uses six function evaluations per step, there's more work per step than when using Euler's method, which has only one function evaluation per step. This work does pay off, as the Fehlberg method is of higher accuracy than Euler's method. The Euler method is very crude due to the local error scaling with h instead of a higher power. (Keeling & Rohani, 2007, p. 25). This means that not as small of a step size must be chosen to achieve a certain accuracy (Zeigler, Muzy, & Kofman, 2018).

5.3 Bogacki-Shapmine Method

Just like Fehlberg's method, the **Bogacki-Shapmine method**, implemented as the ode23 method in software such as MATLAB and R, uses intermediate steps before taking a full step h to further a solution. The method was developed by Przemslaw Bogacki and Lawrence F. Shapmine in a 1989 paper (Bogacki & Shapmine, 1989). The name ode23 indicates that two formulas, one of order 2 and another of order 3 are involved. The 3rd order method is used to advance the solution, while the 2nd order method is used for local error estimation of the second order solution, which is used to adapt the step size to certain error constraints (Moler, 2014).

One step of the Bogacki-Shapmine method uses three stages per step but calculates 4 slopes. This is because it employs a FSAL (first same as last) strategy; k_1 is k_4 from the previous step (Moler, 2014).

The four ks are given by Bogacki's and Shapmine's 1989 paper and are defined in eq. 15.1-15.6.

$$k_1 = f(t_n, y_n), \quad (15.1)$$

$$k_2 = f\left(t_n + \frac{1}{2}h_n, y_n + \frac{1}{2}h_n k_1\right), \quad (15.2)$$

$$k_3 = f\left(t_n + \frac{3}{4}h_n, y_n + \frac{3}{4}h_n k_2\right), \quad (15.3)$$

$$y_{n+1} = y_n + \frac{2}{9}h_n k_1 + \frac{1}{3}h_n k_2 + \frac{4}{9}h_n k_3, \quad (15.4)$$

$$k_4 = f(t_n + h_n, y_{n+1}), \quad (15.5)$$

$$z_{n+1} = y_n + \frac{7}{24}h_n k_1 + \frac{1}{4}h_n k_2 + \frac{1}{3}h_n k_3 + \frac{1}{8}h_n k_4. \quad (15.6)$$

Above, eq. 15.6 is the second order approximation to the exact solution and eq. 15.4 the third order approximation. Because the Bogacki-Shapmine method is of lower order than the Fehlberg method, it is more suitable if only a crude approximation of the solution is sufficient due to there being less work per step. Bogacki and Shapmine argued that their method outperforms other third order methods because of the embedded order two method (Bogacki & Shapmine, 1989)

5.4 Errors and Runge Kutta methods

The local truncation error is the difference between the computed result and the value of the true solution through the previously computed result (Hull, Enright, Fellen, & Sedgwick, 1972). Denoting the results produced by a method by (x_1, y_1) , (x_2, y_2) , (x_3, y_3) ...and denoting the true

solution of the differential equation through (x_{i-1}, y_{i-1}) by (x, x_{i-1}, y_{i-1}) the local error, ignoring round-off errors, is given by $|y_1 - y(x, x_{i-1}, y_{i-1})|$, as seen in figure 9.

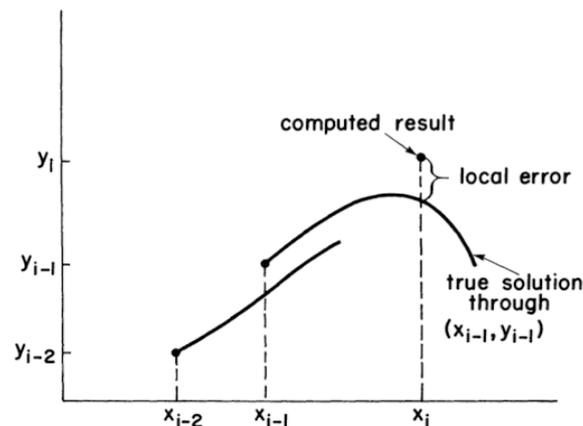


Figure 9: Visual representation of the local truncation error (Hull, Enright, Fellen, & Sedgwick, 1972).

Alongside the local error exists the global error, which describes the accumulation of errors. It is defined as the difference between the computed solution and the real solution. (Shampine & Watts, 1976). If the global error grows without bounds, e.g., the numerical solution rapidly diverges from the true solution, the method is numerically unstable (Stull, 2020).

As we have seen, adaptive step methods have embedded formulas for estimating the truncation error per step. When using software such as MATLAB or R, the user can specify an absolute and relative error tolerance. The absolute error is introduced when the solution nears zero. If the estimated local error exceeds the predetermined error tolerance, the step size is decreased (Hull, Enright, Fellen, & Sedgwick, 1972).

The standard numerical methods introduced so far can be derived by taking finitely many terms of a Taylor series⁶. The Euler method for example, corresponds to the first two terms of a Taylor series. The terms that are omitted during this process constitute the truncation error, and for many methods, the size of this truncation error depends on the step size h (Higham, 2002, S. 5).

The meaning of the term “order” when describing a numerical method, for example the Fehlberg method being of order 4 or Euler method of order 1, describes the way in which the global error behaves. A method being of order p is denoted using “big O notation”. One would write $e_{\text{global}} = O(h^p)$. This means that the global error has the same growth rate as the function h^p (MIT, 2010). The local error on the other hand has an order of $p+1$. A proof of this is laid out

⁶ A Taylor series is a series that uses derivatives to obtain an estimate of a function.

in Chapter III.3 of *Solving Ordinary Differential Equation I* (Hairer, Norsett, & Wanner, 2008). The proof concludes that if the Runge Kutta method is of order p , then the local error admits the rigorous bound defined in inequality 16;

$$\|y_i - y(x_i, x_{i-1}, y_{i-1})\| \leq Ch^{p+1} \quad (16)$$

where C depends on the method and h is the time-step.

Other error-sources include round off errors (Higham, 2002, S. 5), which are caused due to computers only being able to represent numbers by a limited number of binary bits (e.g., 32, 64, 128 bits). This means that precision is limited. Excel, for example has a precision for a specified number confined to 15 significant figures.

5.5 Efficiency of Numerical Integration Methods

When it comes to numerical methods, one is primarily interested the efficiency of a solver. Two terms, the theoretical and real efficiency number are defined to represent the efficiency of a solver (Zhang, 2015). The theoretical efficiency number is given by eq, 17;

$$TFN = \frac{1}{FUNE \times E_{\text{relative}}} \quad (17)$$

where TFN is the theoretical efficiency number and FUNE is the number of function evaluations of each solver and E_{relative} the relative error tolerance. TFN represents the computational efforts required for a certain accuracy constraint.

The real efficiency number is defined in eq. 18;

$$RFN = \frac{1}{10^4 \times TPCU \times E_{\text{relative}}} \quad (18)$$

where TPCU represents the actual computational time for the solver.

The higher the efficiency number, the more efficient the numerical method is at approximating a solution.

6 Combining Mathematics and Biology

6.1 Method

The questions chapter 6 seeks to answer practically are

- How do the parameters β and γ affect the shape of the S, I and R curves? What does this mean for health care systems/society?
- How big are the host groups S, I and R at equilibrium depending on R_0 ?
- Does the behavior of local errors of RK-methods coincide with $O(h^{p+1})$? Do local errors decrease with lower step size?
- How does rounding-precision affect the shape of the curves and their accuracy?
- Are RK-methods with lesser function evaluations per step more efficient than ones with a higher number of function evaluations? How do error constraints affect computational cost?

To answer these questions, numerical experiments are conducted. Because of its user friendliness, Excel is used as means simulating the SIR model. Because Excel doesn't include RK-methods embedded in its software, the formulas for each method must be inputted by hand. The statistical Program R's deSolve package, which specializes in solving ODEs using numerical integration methods such as the RK-integrators is used as well. The reason R was chosen, is its user friendliness for beginners and the simplicity of downloading R packages. To better familiarize oneself with coding in R, the R archive CRAN offers up to date and easy to understand manuals on R packages, which explain how functions used within the package are set up.

To analyze the effects of parameter values on the disease dynamics, the R version 4.1.2 is used. I hypothesize that an increase in the basic reproduction number will lead to a higher number of infected at the peak of infection. The framework for the code used in implementing the SIR model in R was derived from the deSolve package manual (Soetaert, Petzoldt, & Setzer) and RPubS (Choisy, 2018), an open platform for publishing R documents and code. Since the goal is to find out how different β and γ values affect the curves, a for loop is used to iterate the process of running the code once with β and another time with γ ranging from 0.5 to 4, in intervals of 0.5 using the ode45 method with absolute and relative error bound of 10^{-8} , for higher accuracy. This means the model is simulated with R_0 values under and above one. The code for the for loop is derived from a blog post (Bolker, 2016). The results are then exported to Excel for plotting and compared to theory. The max number of infected of one parameter set is measured and compared to that of another set and inferences are made. Equilibria

values are obtained using the rootSolve package in R and compared based on size. The code for the equilibria values is derived from *Epidemics-Models and Data using R* (Bjornstad, 2018).

For the error analysis, Excel is used to implement the basic SIR model w/o demography using the ode23 and Euler method and different step sizes. I hypothesize that a lower step size will lead to a smaller local error. For the local error estimation of the ode23 method, both a second and third order solution is obtained, and the absolute value of their difference is taken as the error estimate for the second order solution. This is done for $h=1$ and $h=0.5$. The errors per step for each h value are compared using error quotients and according to theory, should behave according to the $O(h^{p+1})$, where p denotes the order of the method. For the Euler method, local errors are estimated using a micro-time stepping technique; first, a solution using $h=1$ is derived. Then, each value is used as an initial value for a solution using $h=0.001$. This second solution is then compared to the first by taking the absolute value of the difference of both solutions. After using the same micro-time stepping technique for a $h=0.5$ solution, the local error estimates for $h=1$ and $h=0.5$ (local error is average of half-step errors) are compared using error quotients.

The effect of rounding precision is investigated using Euler and ode23 implementations in Excel. After each step, the solution is rounded to 2-, 3-, 7- or 15-digit precision and used for the next step. Then, the solution for each precision-type over 10 days is plotted in the same graph for easier comparison. Using this one can discern at which precision level the solution stabilizes and make inferences.

To compare computational cost and efficiency of different RK-methods, the model is run in R using relative and absolute error tolerances of 10^{-3} , 10^{-6} , 10^{-10} using the ode23, ode45 and 7th order rk78f method with 8th order error estimate. The computational cost for each method is obtained by running diagnostics and the computational time for each solver is acquired. I hypothesize that the smaller the number of total function evaluations of a solver, the more efficiently the solution is estimated. Using this information, both the theoretical and real efficiency numbers outlined in chapter 5.4 are computed and compared for each solver. Using these values, inferences based on the efficiency of the three RK-methods are made.

7 Results

7.1 Changing Parameter Values

In this section, the results for the R simulations with changing parameter sets are presented. First, results for $R_0 > 1$, specifically $\beta=0.5$ and $\gamma=0.7$ are shown. The effective reproduction number is depicted using a secondary axis. The data sets pertaining to the graphs are found in the appendix.

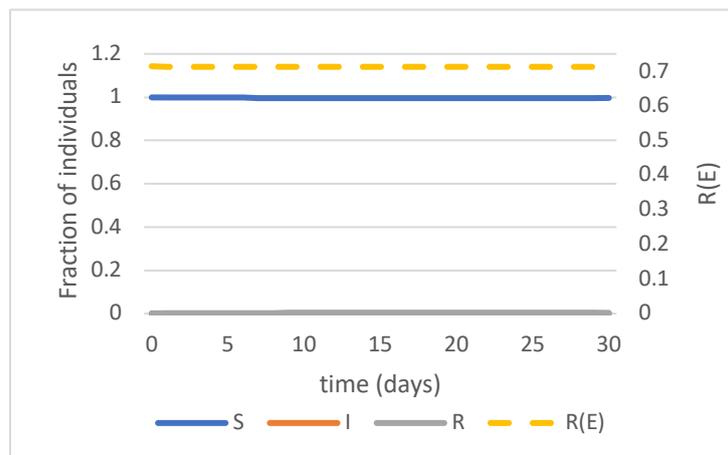


Figure 10: Plot for $\beta=0.5$ and $\gamma=0.7$

Now, for figure 11 both β and γ are increased so that R_0 is just above 1, more specifically $\beta=3$ and $\gamma=2.5$

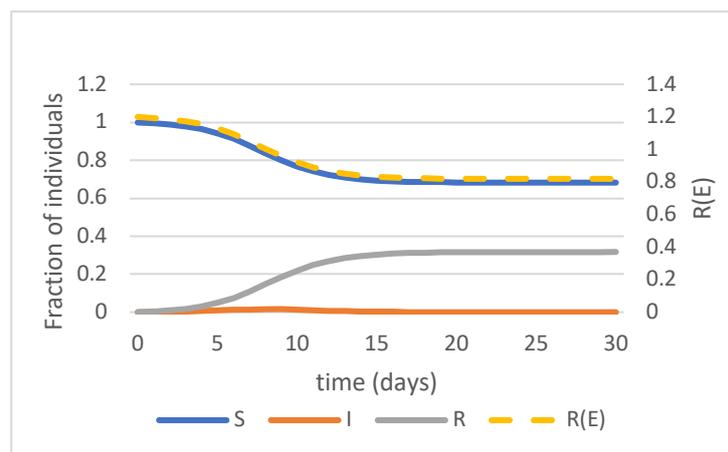


Figure 11: Plot for $\beta=3$ and $\gamma=2.5$

Lastly, the plot for an extreme example for $R_0=5$ is shown in figure 12, where $\beta=3.5$ and $\gamma=0.7$.

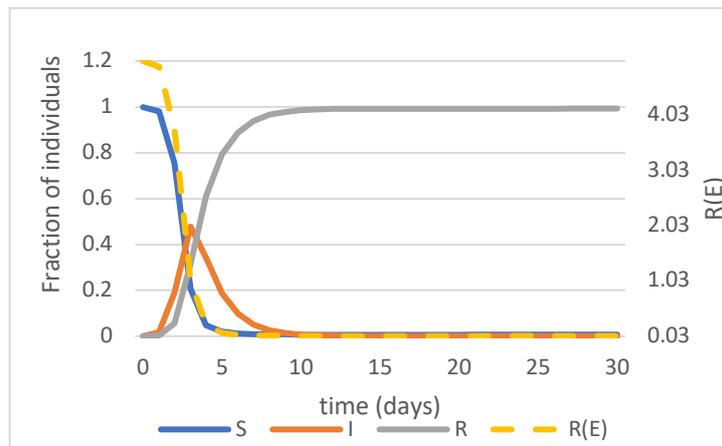


Figure 12: Plot for $\beta=3.5$ and $\gamma=0.7$

7.2 Error Analysis and Rounding

Now, the local error estimates for the Excel ode23 and Euler implementation are presented restricted to 10 days for $\beta=0.3$ and $\gamma=0.2$. The simulations using both methods are found in the appendix.

Table 1: Local error estimates and quotients for ode23 (2nd order solution)

t(d)	Local error est. h=1			Local error est. h=0.5			Quotients		
	S	I	R	S	I	R	S	I	R
1	5.06E-07	1.25E-07	3.82E-07	6.28E-08	1.52E-08	4.76E-08	8.07	8.22	8.02
2	5.29E-07	1.23E-07	4.06E-07	6.54E-08	1.49E-08	5.05E-08	8.08	8.26	8.03
3	5.48E-07	1.18E-07	4.29E-07	6.76E-08	1.42E-08	5.34E-08	8.10	8.33	8.04
4	5.62E-07	1.11E-07	4.51E-07	6.92E-08	1.32E-08	5.61E-08	8.12	8.42	8.05
5	5.70E-07	9.91E-08	4.71E-07	7.01E-08	1.16E-08	5.85E-08	8.14	8.55	8.06
6	5.71E-07	8.31E-08	4.88E-07	7.00E-08	9.46E-09	6.05E-08	8.17	8.79	8.07
7	5.63E-07	6.21E-08	5.01E-07	6.87E-08	6.70E-09	6.20E-08	8.20	9.27	8.08
8	5.45E-07	3.56E-08	5.09E-07	6.62E-08	3.25E-09	6.29E-08	8.24	1.10	8.10
9	5.15E-07	3.24E-09	5.11E-07	6.21E-08	9.44E-10	6.30E-08	8.29	3.43	8.12
10	4.71E-07	3.53E-08	5.06E-07	5.63E-08	5.90E-09	6.22E-08	8.36	5.98	8.14

Table 2: Local error estimates and quotients for Euler

t(d)	Local error est. h=1			Local error est. h=0.5			Quotients		
	S	I	R	S	I	R	S	I	R
1	7.79E-04	4.57E-04	3.21E-04	1.96E-04	1.15E-04	8.11E-05	3.97	3.97	3.96
2	9.87E-04	5.75E-04	4.12E-04	2.52E-04	1.47E-04	1.06E-04	3.91	3.92	3.90
3	1.24E-03	7.15E-04	5.25E-04	3.21E-04	1.84E-04	1.36E-04	3.86	3.87	3.85
4	1.54E-03	8.75E-04	6.66E-04	4.03E-04	2.28E-04	1.75E-04	3.83	3.84	3.80
5	1.89E-03	1.05E-03	8.36E-04	4.96E-04	2.74E-04	2.22E-04	3.80	3.83	3.76
6	2.27E-03	1.23E-03	1.04E-03	5.98E-04	3.20E-04	2.78E-04	3.79	3.84	3.74
7	2.66E-03	1.39E-03	1.28E-03	6.98E-04	3.56E-04	3.43E-04	3.81	3.90	3.72
8	3.02E-03	1.49E-03	1.54E-03	7.84E-04	3.71E-04	4.13E-04	3.86	4.01	3.73
9	3.30E-03	1.48E-03	1.81E-03	8.33E-04	3.50E-04	4.83E-04	3.96	4.23	3.75
10	3.39E-03	1.32E-03	2.08E-03	8.24E-04	2.79E-04	5.45E-04	4.12	4.73	3.81

The quotients for both the ode23 and Euler method are illustrated graphically in figure 13-15 and pertain to the original simulation over 30 days.

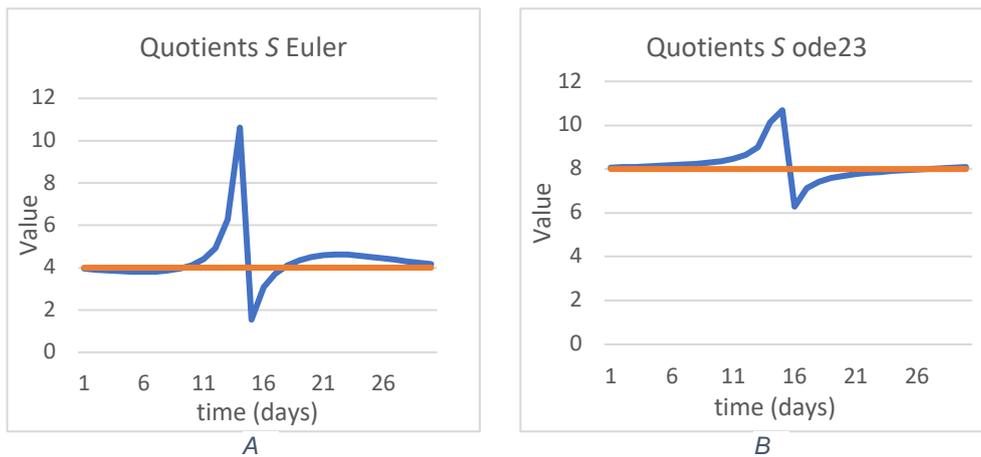


Figure 14: A and B show the plots of the error quotients for the Euler and ode23 solution for S.



Figure 13: A and B show the plots of the error quotients for the Euler and ode23 solution for I.

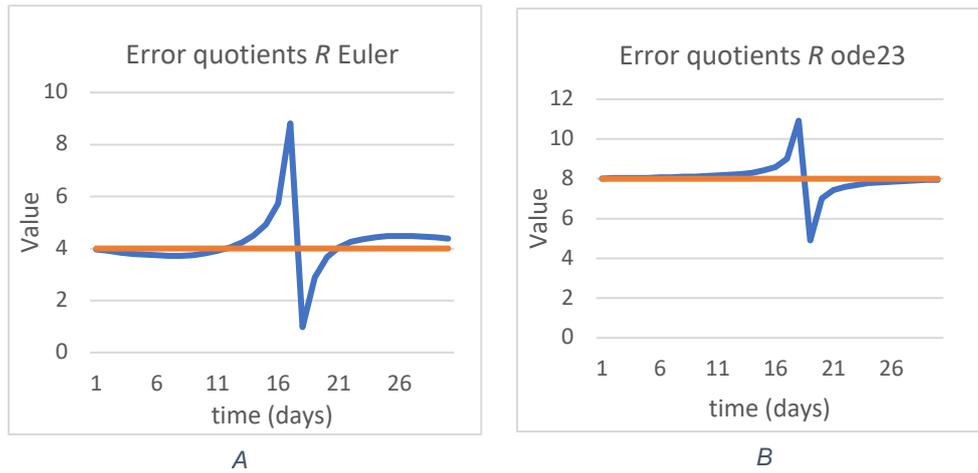


Figure 15: A and B show the plots of the error quotients for the Euler and ode23 solution for R.

Now, the results for the experiments regarding precision are shown in figures 16-18. The simulation was run for both the Euler and ode23 solver to 15-, 7-, 3- and 2-digit precision for $\beta = 0.3$ and $\gamma = 0.2$, restricted to the first 10 days.

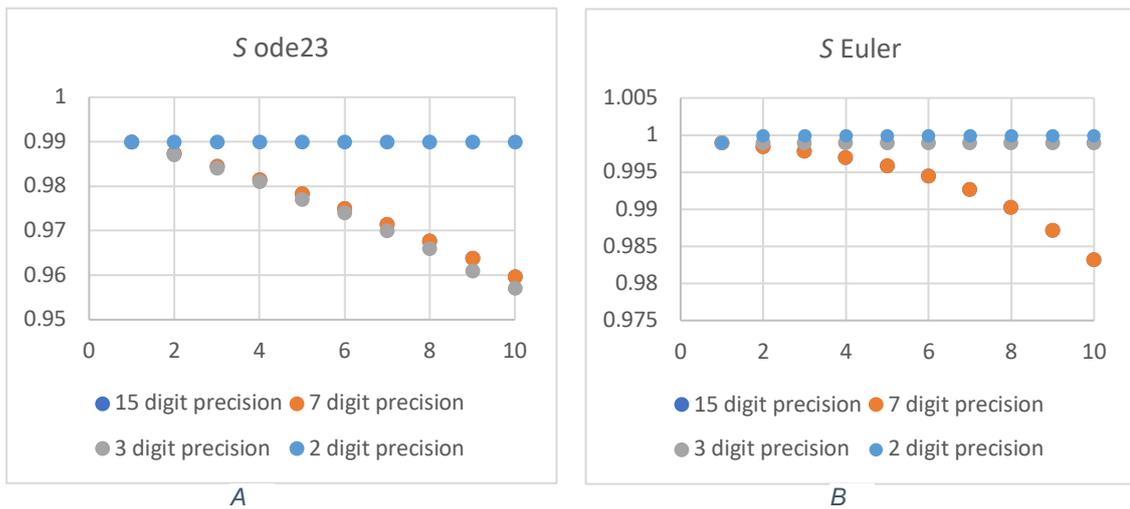


Figure 17: A and B show ode23 and Euler S-curve solutions with different precision levels.

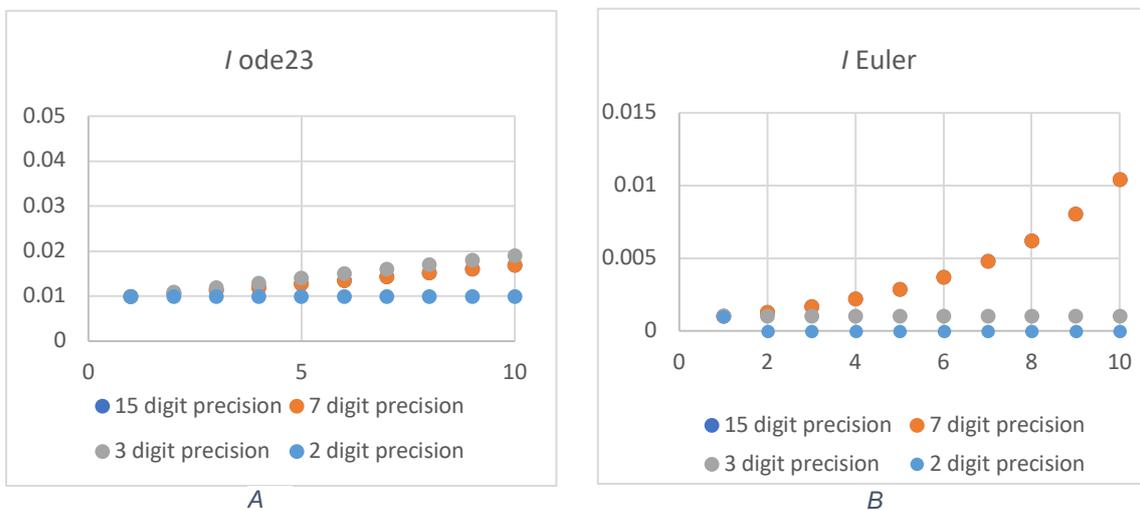


Figure 16: A and B show ode23 and Euler I-curve solutions with different precision levels.

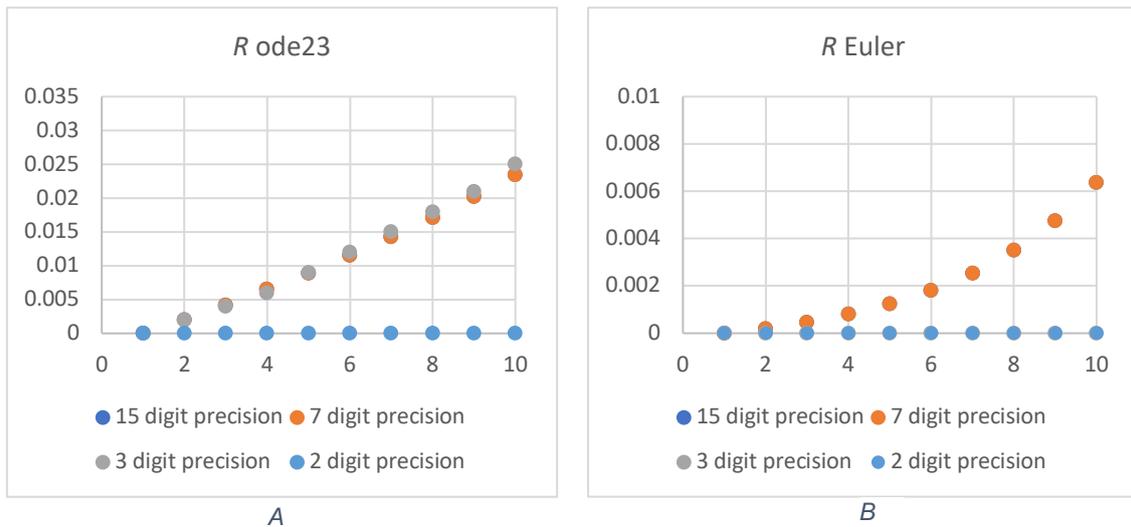


Figure 18: A and B show ode23 and Euler R-curve solutions with different precision levels.

7.3 Efficiency Numbers

In this section, the efficiency and reliability numbers are presented using computational cost and time for the ode23, ode45 and rk78f solutions of the basic SIR model without demography over 30 days with $\beta = 0.3$ and $\gamma = 0.2$. First, the computational values for the error restraint of 10^{-3} are presented in table 3.

Table 3: a)-c) pertain to the diagnostics for relative and absolute error tolerances set to 10^{-3}

a) ode23

number of steps	30
number of function evaluations	92
number of error test failures	0
computational time (s)	0.001822
TFN	1.09E+01
RFN	5.49E+01

b) ode45

number of steps	30
number of function evaluations	182
number of error test failure	0
computational time (s)	0.004927874
TFN	5.49E+00
RFN	2.03E+01

c) rk78f

number of steps	30
number of function evaluations	391
number of error test failures	0
computational time (s)	0.00824285
TFN	2.56E+00
RFN	1.21+01

Now results for an error tolerance of 10^{-6} are presented in table 4.

Table 4: a)-c) pertain to the diagnostics for relative and absolute error tolerances set to 10^{-6}

a) ode23

number of steps	135
number of function evaluations	446
number of error test failures	39
computational time (s)	0.0214808
TFN	2.24E+03
RFN	4.66E+03

b) ode45

number of steps	30
number of function evaluations	182
number of error test failures	0
computational time (s)	0.004426003
TFN	5.49E+03
RFN	2.26E+04

c) rk78f

number of steps	30
number of function evaluations	391
number of error test failures	0
computational time (s)	0.008924961
TFN	2.56E+03
RFN	1.12E+04

Lastly, the results for an error constraint of 10^{-10} are presented in table 5.

Table 5: a)-c) pertain to the diagnostics for relative and absolute error tolerances set to 10^{-10}

a) ode23

number of steps	1514
number of function evaluations	4654
number of error test failures	110
computational time	0.0749811
TFN	2.15E+06
RFN	1.33E+07

b) ode45

number of steps	82
number of function evaluations	496
number of error test failures	2
computational time (s)	0.01651287
TFN	2.02E+07
RFN	6.06E+07

c) rk78f

number of steps	30
number of function evaluations	391
number of error test failures	0
computational time (s)	0.008145094
TFN	2.56E+07
RFN	1.23E+08

8 Discussion

8.1 The Effect of Changing Parametrization

When one compares figures 10-12, the way the infectious curve emerges and grows as the parameter values change becomes notable. In figure 10, all curves are stagnant. This is due to the transmission rate being smaller than the recovery rate ($0.5 < 0.7$); more people leave the infectious class on average per time than enter. This leads to the infection not being able to proliferate, as seen by the flat infectious curve. Consequentially, the disease-free equilibrium is stable ($R_0=0.7$). Comparing the equilibrium values of all three host classes to the theory laid out in chapter 3.2, we can see that they approximately coincide with $(S_E, I_E, R_E) = (1, 0, 0)$.

For figure 11, the curves show a slight change in curvature. Over the first 10 days, roughly 1/5 of the population made up of 1000 individuals leave the susceptible class and move to the infectious class, while a little more than 1/10 of the populace enter the recovered compartment. The maximum number of infected is reached after 8 days at roughly 0.015 (relative value) infected. The small amount of infected at the disease's height is due to the transmission rate being just above the recovery rate and therefore R_0 being larger but close to 1 ($R_0=1.2$). The effective reproduction number at the turnover point (maximal value of infected) is approximately 1, which coincides with the theory laid out in chapter 3.2; the infection proliferates until $R(E)=1$, dying down to 0 and becoming endemic. Because $R_0 > 1$, the endemic equilibrium is stable and shows that the infection dies out because of the exhaustion of the infectious class, and not due to there not being any susceptible left.

Figure 12 shows an extreme example where $R_0=5$. This means the disease spreads rapidly, which is reflected in the maximum of infected at a little less than half the population infected being reached after only 3 days. This confirms the hypothesis that a higher R_0 leads to a higher maximal number of infected. Because the basic reproduction number is so high, virtually all susceptible become infected, as can be seen in the number of susceptible at the stable endemic equilibrium. After reaching its height ($R(E)=1$), the infectious curve dies down to 0, at which point approximately the total population is comprised of recovered individuals. It is also noteworthy that for each simulation, the total population is conservative, as described in the theory. This indicates that the errors from the simulation are well under control, as the total population always stays at 1.

As seen in figures 10-12, the larger the basic reproduction number, the faster the disease proliferates and the higher the maximal amount of infected. This has substantial implications

for healthcare systems; If the spread of a disease isn't mitigated, the rapid nature of its spread can overwhelm health infrastructure and cause a shortage in supplies such as hospital beds or protective gear. To prevent or mitigate this, disease-prevention strategies such as quarantine, handwashing, mask-wearing, social distancing, medicine, and vaccination campaigns are employed. The effect of these measures is a flattening of the curve as either the recovery rate is increased so that it is larger than the transmission rate, or the transmission rate is decreased so that it becomes smaller than the recovery rate, in both cases effectively decreasing R_0 so that it is under 1.

8.2 Error Analysis and the Effect of Rounding

Looking at tables 1 and 2, a clear trend shows itself, the smaller the chosen time step h , the smaller the local error and. This validates my hypothesis laid out earlier, that a smaller h leads to a smaller local error estimate. Quantifying the rate at which the local error decreases is laid out theoretically in chapter 5.2; for the Euler method, the local error scales with h^2 (first order method) and for the ode23 method with h^3 (second order method). This means that a halving of the timestep $h=1$ would lead to the local error decreasing by factor of 4 for Euler's method, and by a factor of 8 for the ode23 method. Comparing this to the error quotients in tables 4 and 5, we can see that this is true. For the ode23 method, the error quotient fluctuates, with some odd spikes which this paper will not delve into further, around 8 and for the Euler method around 4, as is seen in figures 13-15.

Focusing on figures 16-18, it becomes clear that the ode23 solution demonstrates a "stability region" for precision; a precision of 3 digits and larger approximately yields the same solution (7-digit precision solution covers 15-digit precision solution), whereas a 2-digit precision yields a stagnant solution which diverges from the real solution, making it less accurate. The Euler method demonstrates a similar, but smaller, stability region; a precision of 4 digits and larger approximately yield the same solution. This tells us that the Euler method isn't as "resistant" to low precision as the ode23 method. I suspect this is due to the ode23 solution showing a faster initial decrease in susceptible and increase in infected and recovered. If the precision is within the stability regions, precision has a negligible effect on accuracy. Because the global error contains the local error and other errors such as ones due to precision, one can heuristically say that if the local errors do not propagate and the precision is within the stability region the global error doesn't propagate unboundedly either, e.g., the solution is numerically stable.

8.3 Analysis of Efficiency numbers

When it comes to computational cost, tables 3 tell us that for a solution with a relative and absolute error tolerance of 10^{-3} all three solutions yield the same amount of steps, but focusing on function evaluations the ode23 method shows the smallest value. This is because compared to the fourth order ode45 and seventh order rk78f method it only has 3 function evaluations per step, which is reflected in there being roughly 3 times the amount of total function evaluations than steps. The ode45 and rk78f method have 6 and 13 function evaluations per step respectively.

Focusing on computational time, the ode23 solution took the least amount of time to compute at approximately 0.0018s, whereas the other two solutions took more than double that. This can be explained using the total number of function evaluations per method; the higher the order, the more function evaluations per step and the longer it takes to approximate solutions of crude error tolerance. The number of error test failures being 0 for all three methods is noteworthy; this is most likely due to the low error tolerance of 10^{-3} , and the “right” step size, which isn’t very small, being chosen at the start of each step.

When it comes to the TFN and RFN values, we can see that the ode23 method ranks first in both categories, meaning that for a crude error tolerance, a low order method is the most efficient.

Moving on to table 4, the lower error tolerance of 10^{-6} had the effect of increasing the computational time for the ode23 solver but keeping it roughly the same for the ode45 and rk78f method, the ode45 method being the fastest of all three. For both ode45 and rk78f, the number of steps and function evaluations didn’t change, whereas for the ode23 method they increased to 135 and 446 respectively. This is due to higher order methods being able to stay within lower error tolerances without needing to compromise computational cost. It is also noteworthy that the ode23 method produced the most error test failures, probably due to the same reason. The other two methods produce 0.

Comparing the TFN and RFN of each method, the ode45 method is the most efficient at estimating the solution. The ode23 method comes in last place for both metrics and the rk78f method in second. This is because the ode45 solution produced both the smallest computational time and number of function evaluations.

When looking at the results for an absolute/relative error tolerance of and 10^{-10} , we can see that for high accuracy constraints, the rk78f method is the most efficient, the ode45 method coming in second and ode23 method in last place.

The computational cost (number of steps and function evaluations) did not change for the rk78f method, whereas for the ode23 and ode45 method it increased, the former more drastically than the latter (approximately factor of 10 vs. factor of 3). This can be explained by the ode45 method being of one order higher than the ode23 method and this having the effect of the ode45 method being able fulfill the same error tolerance with a larger step size. It is noteworthy that for all error constraints, the computational time for the 7th order method stayed virtually the same, which is due to the computational cost also staying the same. This means that the higher the order, the less computational cost needs to grow for higher accuracy constraints. The number error test failures increased approximately by a factor of 3 for the ode23 method and rose from 0 to 2 for the ode45 method. The number of error test failures stayed at 0 for the rk78f solution.

Methods with a low number of function evaluations per step do not always behave most efficiently, as the number of function evaluations per step is proportional to order, which increases with the accuracy of a method. This has the effect of low order methods needing to take very small steps to stay within predetermined accuracy constraints and thus increasing the total amount of function evaluations (more computational cost) and computational time, leading to lower efficiency. For crude error tolerances however, low order methods, such as the ode23 method, proved most efficient. In each example, the method with least amount of total function evaluations proved most efficient in terms of TFN and RFN, thus proving my hypothesis. This is due to computational time decreasing with a lower total number of function evaluations and this quotient of these two values being used for the efficiency numbers.

9 Conclusion

We gave an answer to how the model parameters affect the S, I and R curves; the parameters β and γ affect the disease dynamics by either flattening the infectious curve and thus leading to a less rapid decrease in susceptible or a heightening of its spread. According to theory, the basic reproduction number depends on the parameters and indicates how fast a disease proliferates. If $R_0 < 1$, the equilibrium values are low, if $R_0 > 1$, almost all the susceptible become infected. My hypothesis that a larger basic reproduction number leads to a higher maximal number infected was confirmed. The way in which R_0 affects the equilibrium became clear; if the basic reproduction number is high, virtually everyone becomes infected and the population at endemic equilibrium will be comprised of recovered individuals. If it is under 1, the disease cannot spread, and the endemic equilibrium is stable. How this affects healthcare systems and society was laid out; the higher R_0 , the higher the threat of the healthcare system being overwhelmed. R_0 can be decreased using disease prevention and mitigation strategies.

We have seen that the local errors of the Euler and ode23 method do scale with $O(h^{p+1})$, which means that a smaller step size leads to a smaller local error. This answers the question of whether error size decreases with step size and confirms my hypothesis. Both methods yield similar precision stability regions; the ode23 method yielded approximately the same solution for 3-digit precision and higher, while the Euler solution stayed the same for 4-digit precision and higher. For the ode23 method, intermediately rounding the solution to 2-digits meant that all 3 curves remained stagnant. The same thing happened for the Euler solution for a 3-digit precision and lower. If the precision is within the established stability regions, the global error can be controlled, at least heuristically, using the local error as the error due to rounding is small. This gives us insight into how precision affects the accuracy and shape of the solution curves.

Regarding efficiency, the ode23 method proved most efficient in terms of both theoretical and real efficiency and demonstrated least amount of total function evaluations and shortest computational time for a relative and absolute error constraint of 10^{-3} . The ode45 method proved most efficient for an absolute and relative error constraint of 10^{-6} , and the rk78f method for a low error tolerance of 10^{-10} . This means that methods with a lower number of function evaluations per step aren't necessarily more efficient, it depends on how crude or precise the desired accuracy is. If a low accuracy is sufficient, lower order methods yield highest efficiency and least computational cost. The opposite holds for high accuracy constraints. This gives an answer to whether methods with a low number of function evaluations per step are more efficient. For each example, the method with the smallest number of total function evaluations

proved the most efficient, confirming my hypothesis. The efficiency experiments showed that the error constraint affects the computational cost of low order methods more than for higher order methods. For low order methods, a moderately higher accuracy constraint increases the computational cost, whereas for high order methods it stays the same, which gives us an answer to how error constraints affect computational cost.

As an extension to the research outlined in this paper, focus could be shifted to implicit numerical methods, such as the backwards Euler method, regarding accuracy, since certain problems, especially numerical unstable ones, are better suited for such methods. The experiments conducted in this paper could be executed using the SIR model with demography to analyze how birth and death rates effect the dynamics of the curves. One could also study the dynamics for mortality inducing diseases using a more complex SIR model that takes a changing population size into consideration. The odd spikes that occur for the local errors of the second order ode23 and Euler solution could be investigated in depth.

10 Reflection

All in all, the writing of this project was a very positive experience for me. I was able to intensively research a topic I am deeply interested in, whilst collaborating with my math teacher Mr. Preu, whose help and guidance I very much appreciated. Some aspects, such as the usage of R as a novice or the tedious manual implementation of Runge-Kutta formulas in Excel proved challenging, but I am grateful for the insight it gave me into epidemiological modelling.

11 Bibliography

- Ashcroft, P., Bonhoeffer, S., Kouyos, R., Regoes, R. and Stadler, T., (2020). *Infectious disease dynamics*. Spring Semester 2020. Zürich; Eidgenössische Technische Hochschule Zürich.
- Amalia S. Magaret, A. M.-M. (2016). Effect of Condom Use on Per-act HSV-2 Transmission Risk in HIV-1, HSV-2-discordant Couples. *Clinical Infectious Diseases*.
- Bacaer, N. (2011). *A Short History of Mathematical Population Dynamics*. London: Springer.
- Bacaer, N. (2011). <https://healthtian.com/helminths/>.
- Baratt, H., Kirwan, M., & Shantikumar. (2009). Retrieved from healthknowledge.org.uk: <https://www.healthknowledge.org.uk/public-health-textbook/research-methods/1a-epidemiology/epidemic-theory>
- Bjornstad, O. N. (2018). *Epidemics-Models and Data using R*. Springer Nature Switzerland AG.
- Bogacki, P., & Shapmine, L. (1989). A (3)2 Pair of Runge-Kutta Formulas. *Applied Mathematic Letters*.
- Bolker, B. (2016). *Using a loop in an ODE to graphically compare different parameters R*. Retrieved from stackoverflow.com: <https://stackoverflow.com/questions/41324295/using-a-loop-in-an-ode-to-graphically-compare-different-parameters-r>
- Britannica, T. E. (1998). *Fluke*. Retrieved from Britannica.com: <https://www.britannica.com/animal/fluke-flatworm>
- Britannica, T. E. (1998). *Daniel Bernoulli*. Retrieved from Britannica.com: <https://www.britannica.com/biography/Daniel-Bernoulli>
- Britannica, T. E. (1998). *Sir Ronald Ross*. Retrieved from Britannica.com: <https://www.britannica.com/biography/Ronald-Ross>
- Choisy, M. (2018). SIR Models in R.
- Cleaveland, S., Laurensen, M., & Taylor, L. (2001, 7 29). Diseases of humans and their domestic mammal: pathogen characteristics, host range and the risk of emergence. *Philos Trans R Soc Land B Biol Sci*, 991-999.
- Dadlani, A., Richard, O., Afolabi, Jung, H., Sohraby, K., & Kim, K. (2013). *Deterministic Models in Epidemiology: From Modeling to Implementation*.

- Technical Report, Gwangju Institute of Science and Technology, communications and Sensor Networks laboratory, Gwangju, South Korea.
- Darling, D. (2016). *deterministic system*. Retrieved from [daviddarling.info](http://www.daviddarling.info/):
http://www.daviddarling.info/encyclopedia/D/deterministic_system.html
- Fehlberg, E. (1969). *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*. NASA.
- Hairer, E., Norsett, S., & Wanner, G. (2008). *Solving Ordinary Differential Equations I*. Springer.
- Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms*.
- Hoffmann, Brindley, & Berriman. (2014). *Halting harmful Helminths*. Retrieved from Science: <https://healthtian.com/helminths/>
- Hull, Enright, Fellen, & Sedgwick. (1972). *Comparing Numerical Methods for Ordinary Differential Equations*. Society for Industrial and Applied Mathematics.
- Keeling, M., & Rohani, P. (2007). *Modeling Infectious Diseases and Humans and Animals*. Princeton University Press.
- Kong, Q., Siau, T., & Bayen, A. M. (2020). *Python Programming and Numerical Methods*. Academic Press.
- Mahaffy, J. M. (2018). Math 363- Mathematical Modeling.
- MIT. (2010). Big O notation.
- Moler, C. (2014, May 26). *Ordinary Differential Equation Solvers ODE23 and ODE45*. Retrieved from MathWorks:
<https://blogs.mathworks.com/cleve/2014/05/26/ordinary-differential-equation-solvers-ode23-and-ode45/>
- Parker, G., Ball, M., & Chubb, J. (2015, 2 28). Evolution of complex life cycles in tropically transmitted helminths. *J Evol Biol*, 1(28), 267-291.
- Ross, R. (1917). *Theory of Happenings*.
- Shampine, L., & Watts, H. (1976). *Global Error Estimation for Ordinary Differential Equations*. Sandia Laboratories.
- Soetaert, K., Petzoldt, T., & Setzer, W. (n.d.). Package deSolve: Solving Initial Value Differential Equations in R.
- Stull, R. (2020). *Numerical Errors and Instability*. Retrieved from LibreTexts:
https://geo.libretexts.org/Bookshelves/Meteorology_and_Climate_Science/Boo

- k%3A_Practical_Meteorology_(Stull)/20%3A_Numerical_Weather_Prediction_(NWP)/20.03%3A_Section_4-
- UCLA. (2020). *Learning Epidemic Models for COVID-19*. Retrieved from covid19.uclaml.org: <https://covid19.uclaml.org/model.html>
- Wai, Choon, & Idris. (2021). Adaptive Time-Stepping for Runge-Kutta methods for Ordinary Differential Equations. *Universiti Malaysia Terengganu Journal of Undergraduate Research*.
- WHO. (2022). *HIV*. Retrieved from who.int: <https://www.who.int/data/gho/data/themes/hiv-aids>
- Zeigler, B. P., Muzy, A., & Kofman, E. (2018). *Theory of Modeling and Simulations*. Academic Press.
- Zhang, L. (2015). *Studying and Comparing Numerical Methods for Ordinary Differential Equations*. Massachusetts Institute of Technology.

12 Table of Figures

Figure on title page: provided by canva.com

Figure 1: Portrait of Daniel Bernoulli (Britannica T. E., 1998)	6
Figure 2: Ronald Ross, Anderson G. McKendrick, and William O. Kermack (Britannica T. E., 1998), (Bacaer N., 2011)	7
Figure 3: Helminths and flukes (Hoffmann, Brindley, & Berriman, 2014), (Britannica, 1998)	9
Figure 4: The movement of individuals from one host group to the other (UCLA, 2020).	11
Figure 5: Numerical simulation of an Influenza outbreak in R, over 30 days (based on Mahaffy, 2018)	14
Figure 6: Numerical simulation of SI model herpes outbreak in R, over 10'000 days	17
Figure 7: Numerical Simulation of SEIR model in R	18
Figure 8: Coefficients of Fehlberg's formulas	20
Figure 9: Visual representation of the local truncation error (Hull, Enright, Fellen, & Sedgwick, 1972).	22
Figure 10: Plot for $\beta=0.5$ and $\gamma=0.7$	26
Figure 11: Plot for $\beta=3$ and $\gamma=2.5$	26
Figure 12: Plot for $\beta=3.5$ and $\gamma=0.7$	27
Figure 14: A and B show the plots of the error quotients for the Euler and ode23 solution for I.	28
Figure 13: A and B show the plots of the error quotients for the Euler and ode23 solution for S.	28
Figure 15: A and B show the plots of the error quotients for the Euler and ode23 solution for R.	29
Figure 17: A and B show ode23 and Euler I-curve solutions with different precision levels.	29
Figure 16: A and B show ode23 and Euler S-curve solutions with different precision levels.	29
Figure 18: A and B show ode23 and Euler R-curve solutions with different precision levels.	30

13 List of Tables

Table 1: Local error estimates and quotients for ode23 (2 nd order solution)	27
Table 2: Local error estimates and quotients for Euler	28
Table 3: a)-c) pertain to the diagnostics for relative and absolute error tolerances set to 10^{-3}	30
Table 4: a)-c) pertain to the diagnostics for relative and absolute error tolerances set to 10^{-6}	31
Table 5: a)-c) pertain to the diagnostics for relative and absolute error tolerances set to 10^{-10}	32

14 Declaration

Erklärung

„Ich erkläre, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Alle wörtlichen und sinngemässen Übernahmen aus andern Werken habe ich als solche kenntlich gemacht.

Ich nehme ausserdem zur Kenntnis, dass meine Arbeit zur Überprüfung der korrekten und vollständigen Angabe der Quellen mit Hilfe einer Software (Plagiaterkennungstool) geprüft wird.“

Datum 3.12.2022

Unterschrift *Ariana Neumann*

15 Work Journal

October of 2021	1st introduction to topic via Mr. Preu.
November 2021 - January 2022	Preliminary research, gathering of information regarding SIR models and potential implementation.
January	First time trying R, specifically implementing basic SIR model (setting of initial values, SIR equations, parameter values). This constituted preliminary experimenting.
February	Research into numerical integration methods such as Euler using material provided by Mr. Preu.
March- April	Research into implementing RK-methods in R for SIR model and different types of SIR models (with and without vital dynamics).
May 18th	1st meeting with Mr. Preu setting down goals of the paper and implementation methods of the SIR model (R and Excel) and filling out of contract.
May 25th	Signing of contract
June 8th	<p>Meeting with Mr. Preu; Mr. Preu and I discussed potential questions for my meeting with Prof. Kuoyos on the 14th of June, and agreed on formulating most of the questions in a more general matter, only going into specifics if necessary. We also looked at ETH lecture notes provided by Prof. Stadler regarding infectious disease dynamics, and discussed integration of a logistic differential equation regarding an analytical solution to the SI model, and agreed that I will not be employing such analytical methods, as they are beyond my scope. I will be using numerical integration methods instead.</p> <p>Mr. Preu provided me with a book on Modeling infectious diseases written by Matt J. Keeling and Pejman Rohani, which will help me in gathering real life data and their associated graphs, which I can directly download. We agreed to meet again on June 16th 2022.</p>
June 14th	<p>Meeting with Prof. Kuyos; I was able to meet with Professor Kuyos who leads a Kohort on HIV research. He was not only able to convey useful tips when modelling infectious diseases (such as structuring your differential equations according to the disease or making me aware of the fact that the euler method tend to become downwards unstable when I converges to 0) but also reccomend a book on infectious disease modelling in R. I asked him questions regarding what to look out for when modelling dieseases, stochastic models, error mitigation and data gathering. He was able to grant me insight into the disease modelling process by explaining to me the programs he uses to run simulations (R and C++) and parameter "fitting", a process in which he compares real life data points to his simulation in order to discern the values of the model parameters.</p>
June 16th	<p>Meeting with Mr. Preu; We discussed the meeting with Prof. Kuyos and I explained the various pieces of information he was able to convey. I also showed Mr. Preu some pages of the book he recommended to me. He asked me what I envisioned my next steps to be regarding my project, and we agreed that I would start by</p>

	simulating the basic SIR model in Excel and R and playing around with timestepping values, rounding and parameter values (will be using parameter values from real life data if possible), and including the basic and effective reproduction number in my simulation and endemic equilibria calculations.
July 6th	Meeting with Mr. Preu; We discussed potential ways to implement error estimations in my project. We agreed that I'd try to implement the ode23 method in Excel, export the yielded data into R and use it as initial values in the ode45 method and calculate the error by comparing my Excel ode23 solution with the R ode45 solution at certain time steps. We looked at local errors with the help of two papers (technical data report on numerical integration methods published by MIT and SIAM paper). The former paper included a formula used to estimate the efficiency of a numerical integration method.
July 7th	1st time implementation of ode 23 method in Excel
August 1st	Creation of preliminary table of contents in order to be more efficient when drafting content of paper
August 7th	Running Simulation basic SIR model in R; I first experimented with parameter values (β ranging from 0.5 to 4 and $\gamma=0.7$ and γ ranging from 0.5 to 4 and $\beta=3$) using a for-loop and the ode45 numerical integration method, plotting the output subsequently. I then saved the output into an Excel document for easier access.
August 8th	Implementing ode 23 and estimating error using difference of third order and second order method in Excel as an alternative to proposed method because of failure to do so in R. Simulating herpes using SI model and simulating SEIR model, and SIR model including vital dynamics in R. Implementation of ode23, ode45, rk78f in R in order to find out the number of steps needed depending on error tolerance so that I can calculate efficiency numbers in the future.
August 9th	Drafting of preliminary introduction
August 23rd	Meeting with Mr Preu; We discussed my preliminary table of contents and he suggested I separate the section concerning mathematical methods from the more biological/epidemiological theory sections and made minor changes regarding language. We also briefly spoke about APA-style citing and decided that the paper would adhere to it. I asked him a question regarding my ode23 Excel implementation, because I realized that my local truncation error estimations do not behave in the way that its respective big O notation describes it should.
August 29th	Drafting of Preamble
August 30th	Implementing the Euler method in Excel, local error estimation using timestepping, validation of $O(h^2)$ order of the euler method by calculating error quotients for $h=1$ and $h=0.5$
September 5th	Drafting of first part of theory pertaining to types of diseases
September 11th	Drafting of theory part regarding deterministic disease modelling and the SIR differential equations. Writing about how the parameters are defined and the transmission term is derived and why one distinguishes between frequency and density dependent transmission.

September 19th	Impromptu meeting with Preu. We looked over the Excel sheets.
October 2nd	Drafting of theory part regarding the basic reproduction number and extensions of the SIR model (SEIR and SI). Writing about derivation of R_0 for the different models.
October 10th	Correction of ode23 Excel implementation due to typing mistakes in formula, yielding of correct $O(h^3)$ order by calculating error quotients for $h=1$, $h=0.5$ and $h=0.25$
October 18th	Writing of theory part concerning mathematical methods, specifically explaining the Runge Kutta methods (Fehlberg and Bogacki-Shampine). Explanation of methods used to measure efficiency and reliability of a method. Explanation of error behaviour and order of a method.
October 20th	Drafting of method chapter
October 21st	Compiling of graphs and tables used in results chapter, Analysis of findings and drafting of conclusion
October 22nd	Drafting of abstract
November 6th- December 8th	Editing and formatting of project
December 12th	Handing in of project

Appendix I

R-code

```

#Code for changing  $\beta$  values, ranging from  $\beta = 0.5$  to  $\beta = 4$  by half steps and  $\gamma = 0.7$ 
library(deSolve)
require(rootSolve)
library(rio)
SIR <- function(time,state,parameters) {
  with(as.list(c(state,parameters)),{
    dS <-  $-\beta * S * I$ 
    dI <-  $\beta * S * I - \gamma * I$ 
    dR <-  $\gamma * I$ 
    return(list(c(dS,dI,dR)))
  })
}
init <- c(S=0.999, I=0.001, R=0)
times <- seq(0,30,by=1)
 $\beta$  vec <- seq(0.5,4,by=0.5)
res<- vector(length( $\beta$  vec),mode="list")
library(deSolve)
for (k in seq_along( $\beta$  vec)){
  res[[k]] <- rk(y=init,times=times,func=SIR,
    parms=c( $\beta = \beta$  vec[k],  $\gamma = 0.7$ ), method=rkMethod("ode45"), rtol=1e-8, atol=1e-
8)
}

#exporting data into Excel
export(res, "parameter_runs_with_ $\beta$  _changing.xlsx")

#getting equilibria of each run
equil1=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta =$ 
0.5,  $\gamma = 0.7$ ))
equil2=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta = 1,$ 
 $\gamma = 0.7$ ))
equil3=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta =$ 
1.5,  $\gamma = 0.7$ ))
equil4=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta = 2,$ 
 $\gamma = 0.7$ ))
equil5=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta =$ 
2.5,  $\gamma = 0.7$ ))
equil6=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta = 3,$ 
 $\gamma = 0.7$ ))
equil7=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta =$ 
3.5,  $\gamma = 0.7$ ))
equil8=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c( $\beta = 4,$ 
 $\gamma = 0.7$ ))

#Code for changing  $\gamma$  values, ranging from  $\gamma = 0.5$  to  $\gamma = 4$  by half steps and  $\beta = 3$ 
 $\gamma$ vec <- seq(0.5,4, by=0.5)
res2 <- vector(length( $\gamma$ vec),mode="list")
for (k in seq_along( $\gamma$ vec))
  res2[[k]] <- rk(y=init,times=times,func=SIR,
    parms=c( $\beta = 3, \gamma = \gamma$ vec[k]),method=rkMethod("ode45"), rtol=1e-8, atol=1e-8)

```

```

#exporting data into Excel
export(res2, "parameter_runs_with_γ_changing.xlsx")

#getting equilibria of each run
equil9=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c(β = 3,
γ= 0.5))
equil10=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c(β =
3, γ= 1))
equil11=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c(β =
3, γ= 1.5))
equil12=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c(β =
3, γ= 2))
equil13=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c(β =
3, γ= 2.5))
equil14=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c(β =
3, γ= 3))
equil15=runsteady(y=c(S=0.999, I=0.001, R=0), times = c(0,30), func= SIR, parms=c(β =
3, γ= 3.5))

#Using ode23 and error tolerances to simulate model, with measuring of
computational time
start1 <- Sys.time()
SIR1_rk23 <- ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c(β =0.5,
γ=0.2), method=rkMethod("ode23"), rtol = 1e-3, atol = 1e-3)
end1 <- Sys.time()
time1 <- end1 - start1
time1
start2 <- Sys.time()
SIR2_rk23 <- ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c(β =0.5,
γ=0.2), method=rkMethod("ode23"), rtol = 1e-6, atol = 1e-6)
end2 <- Sys.time()
time2 <- end2 - start2
time2
start3 <- Sys.time()
SIR3_rk23 <- ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c(β =0.5,
γ=0.2), method=rkMethod("ode23"), rtol = 1e-10, atol = 1e-10)
end3<- Sys.time()
time3 <- end3 - start3
time3

#running diagnostics
diagnostics.deSolve(SIR1_rk23)
diagnostics.deSolve(SIR2_rk23)
diagnostics.deSolve(SIR3_rk23)

#Using ode45 and error tolerances to simulate model, with measuring of computational
time
start4 <- Sys.time()
SIR1_rk45 <-ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c(β =0.5,
γ=0.2), method=rkMethod("ode45"), rtol = 1e-3, atol = 1e-3)
end4 <- Sys.time()
time4 <- end4 - start4
time4
start5 <- Sys.time()
SIR2_rk45 <-ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c(β =0.5,
γ=0.2), method=rkMethod("ode45"), rtol = 1e-6, atol = 1e-6)

```

```

end5 <- Sys.time()
time5 <- end5 - start5
time5

start6 <- Sys.time()
SIR3_rk45 <- ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c( $\beta$  =0.5,
 $\gamma$ =0.2), method=rkMethod("ode45"), rtol = 1e-10, atol = 1e-10)
end6 <- Sys.time()
time6 <- end6 - start6
time6

#running diagnostics
diagnostics.deSolve(SIR1_rk45)
diagnostics.deSolve(SIR2_rk45)
diagnostics.deSolve(SIR3_rk45)

#Using rk78f and error tolerances to simulate model, with measuring of computational time
start8 <- Sys.time()
SIR4_rk78f1 <- ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c( $\beta$  =0.5,
 $\gamma$ =0.2), method=rkMethod("rk78f"), rtol = 1e-3, atol = 1e-3)
end8 <- Sys.time()
time8 <- end8 - start8
time8
start8 <- Sys.time()
SIR4_rk78f2 <- ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c( $\beta$  =0.5,
 $\gamma$ =0.2), method=rkMethod("rk78f"), rtol = 1e-6, atol = 1e-6)
end8 <- Sys.time()
time8 <- end8 - start8
time8
start9 <- Sys.time()
SIR4_rk78f3 <- ode(y=c(S=0.999, I=0.001, R=0), times=times, func=SIR, parms=c( $\beta$  =0.5,
 $\gamma$ =0.2), method=rkMethod("rk78f"), rtol = 1e-10, atol = 1e-10)
end9 <- Sys.time()
time9 <- end9 - start9
time9

#running diagnostics
diagnostics.deSolve(SIR4_rk78f1)
diagnostics.deSolve(SIR4_rk78f2)
diagnostics.deSolve(SIR4_rk78f3)

#Coding different infectious models
SI <- function(time,state,parameters) {
  with(as.list(c(state,parameters)),{
    dS <- - $\beta$  *I*S+ $\mu$ -S* $\mu$ 
    dI <-  $\beta$  *I*S- $\mu$ *I
    return(list(c(dS,dI)))
  })
}

SI_model <- ode(y=c(S=0.999, I=0.001), parms=c( $\mu$ =0.001,  $\beta$  =0.0023), func=SI,
method=rkMethod("ode45"), rtol=1e-10, atol=1e-10, times = seq(0,1000,by=100))

SI_model <- as.data.frame(SI_model)

with(SI_model ,{

```

```

    plot(time, S, type="l", col="blue", xlab="time(days)", ylab="fraction of people", ylim=c(0,1))
    lines(time, I, col="green")
  })
  legend("right", legend=c("S", "I"),
        col=c("blue", "green"), lty=1, cex=0.8,
        title="Line types", text.font=4, bg='lightblue')
  SEIR <- function(time,state,parameters) {
    with(as.list(c(state,parameters)),{
      dS <- -β *S*I+ mu - mu*S
      dE <- β *S*I - mu*E - sigma*E
      dI <- -γ*I - mu*I+sigma*E
      dR <- γ*I - mu*R
      return(list(c(dS,dE,dI,dR)))
    })
  }
  SEIR_model <- ode(y=c(S=0.998, E=0.001, I=0.001, R=0 ), parms=c(mu=0.001, β =0.3,
γ=0.2, sigma=0.1), func=SEIR, method=rkMethod("ode45"), rtol=1e-10, atol=1e-10, times =
seq(0,10000,by=100))
  SEIR <- as.data.frame(SEIR_model)
  with(SEIR ,{
    plot(time, S, type="l", col="blue", xlab="time(days)", ylab="fraction of people", ylim=c(0,1))
    lines(time, E, col="magenta")
    lines(time, I, col="green")
    lines(time, R, col="red")
  })
  legend("right", legend=c("S", "E", "I", "R"),
        col=c("blue", "purple", "green", "red"), lty=1, cex=0.8,
        title="Line types", text.font=4, bg='lightblue')

```

Data Parameter Experiments

Below is the compiled data of the experiments with changing parametrization in R, starting with changing transmission rates.

$$\beta=0.5, \gamma=0.5$$

time (days)	S	I	R	R(E)
0	0.999	0.001	0	0.71428571
1	0.99854751	0.000818226	0.00063427	0.71324822
2	0.99817746	0.000669356	0.00115318	0.7129839
3	0.99787486	0.000547481	0.00157765	0.71276776
4	0.99762745	0.000447735	0.00192481	0.71259104
5	0.99742517	0.00036612	0.00220871	0.71244655
6	0.9972598	0.000299356	0.00244084	0.71232843
7	0.99712461	0.000244747	0.00263064	0.71223187
8	0.9970141	0.000200089	0.00278581	0.71215293
9	0.99692377	0.000163571	0.00291266	0.71208841
10	0.99684993	0.000133712	0.00301636	0.71203566
11	0.99678957	0.0001093	0.00310113	0.71199255
12	0.99674024	8.93426E-05	0.00317042	0.71195732
13	0.99669992	7.30277E-05	0.00322705	0.71192851
14	0.99666696	5.96909E-05	0.00327335	0.71190497
15	0.99664002	4.87891E-05	0.00331119	0.71188573
16	0.996618	3.98778E-05	0.00334212	0.71187
17	0.99660001	3.25939E-05	0.0033674	0.71185715
18	0.9965853	2.66402E-05	0.00338806	0.71184664
19	0.99657328	2.17739E-05	0.00340495	0.71183806
20	0.99656345	1.77964E-05	0.00341875	0.71183104
21	0.99655542	1.45454E-05	0.00343003	0.7118253
22	0.99654886	1.18882E-05	0.00343925	0.71182061
23	0.9965435	9.71648E-06	0.00344679	0.71181678
24	0.99653911	7.94144E-06	0.00345295	0.71181365
25	0.99653553	6.49065E-06	0.00345798	0.71181109
26	0.9965326	5.30489E-06	0.0034621	0.711809
27	0.99653021	4.33575E-06	0.00346546	0.71180729
28	0.99652825	3.54366E-06	0.00346821	0.71180589
29	0.99652665	2.89627E-06	0.00347045	0.71180475
30	0.99652534	2.36715E-06	0.00347229	0.71180382

$$\beta = 1, \gamma = 0.7$$

time (days)	S	I	R	R(E)
0	0.999	0.001	0	1.42857143
1	0.99783648	0.00134776	0.00081575	1.42548069
2	0.99627154	0.00181401	0.00191445	1.42324506
3	0.99417101	0.00243711	0.00339188	1.4202443
4	0.99135936	0.00326626	0.00537438	1.41622766
5	0.98760971	0.00436326	0.00802703	1.41087102
6	0.98263361	0.00580347	0.01156292	1.4037623
7	0.97607263	0.00767493	0.01625245	1.39438947
8	0.9674954	0.01007372	0.02243087	1.38213629
9	0.95640584	0.01309345	0.0305007	1.36629406
10	0.94227044	0.01680584	0.04092372	1.34610062
11	0.92457304	0.02123105	0.05419591	1.32081863
12	0.90290234	0.02629939	0.07079827	1.28986048
13	0.87706676	0.03181307	0.09112017	1.25295251
14	0.84721457	0.03742488	0.11536055	1.21030653
15	0.81391806	0.04265538	0.14342656	1.16274008
16	0.77817507	0.04696265	0.17486228	1.11167867
17	0.74130183	0.04985533	0.20884284	1.05900261
18	0.70473539	0.05101194	0.24425267	1.00676484
19	0.66980764	0.05035741	0.27983495	0.95686805
20	0.63756441	0.04806596	0.31436964	0.91080629
21	0.6086758	0.0444959	0.3468283	0.86953686
22	0.58344006	0.0400908	0.37646914	0.83348579
23	0.56185088	0.03528631	0.40286281	0.80264411
24	0.54369016	0.03044716	0.42586268	0.77670022
25	0.52861684	0.02583952	0.44554364	0.75516692
26	0.5162373	0.02163093	0.46213177	0.73748185
27	0.50615353	0.01790616	0.47594031	0.72307648
28	0.49799234	0.0146886	0.48731905	0.71141763
29	0.49142004	0.01196108	0.49661887	0.70202863
30	0.4861478	0.00968274	0.50416946	0.69449685

$$\beta=1.5, \gamma=0.7$$

time (days)	S	I	R	R0(E)
0	0.999	0.001	0	2.14285714
1	0.99671027	0.00221889	0.00107084	2.13580772
2	0.99166155	0.00489776	0.00344069	2.12498903
3	0.98067081	0.01068748	0.0086417	2.10143745
4	0.95739468	0.02275376	0.01985156	2.05156002
5	0.91083574	0.0460479	0.04311636	1.95179086
6	0.82737354	0.08466049	0.08796597	1.7729433
7	0.70286554	0.13305951	0.16407495	1.50614044
8	0.55820336	0.17018197	0.27161467	1.19615007
9	0.42858271	0.17649081	0.39492649	0.91839151
10	0.33349543	0.15451262	0.51199195	0.71463307
11	0.27122814	0.12033444	0.60843742	0.58120316
12	0.23231193	0.08697363	0.68071444	0.49781127
13	0.20825428	0.06001474	0.73173099	0.44625916
14	0.19333313	0.04024153	0.76642533	0.41428529
15	0.18401397	0.02650592	0.78948011	0.39431565
16	0.17815499	0.0172646	0.80458041	0.38176069
17	0.17445244	0.01116633	0.81438123	0.37382665
18	0.17210402	0.00718997	0.82070601	0.36879432
19	0.17061075	0.00461651	0.82477274	0.36559445
20	0.16965964	0.00295882	0.82738154	0.36355638
21	0.1690532	0.00189419	0.82905261	0.36225686
22	0.16866625	0.00121174	0.83012201	0.36142767
23	0.16841923	0.00077481	0.83080596	0.36089835
24	0.1682615	0.00049528	0.83124323	0.36056035
25	0.16816075	0.00031653	0.83152271	0.36034447
26	0.16809641	0.00020227	0.83170132	0.36020658
27	0.1680553	0.00012925	0.83181545	0.3601185
28	0.16802904	8.2582E-05	0.83188838	0.36006223
29	0.16801226	5.2763E-05	0.83193497	0.36002628
30	0.16800155	3.3711E-05	0.83196474	0.36000332

$$\beta=2, \gamma=0.7$$

time (days)	S	I	R	R(E)
0	0.999	0.001	0	2.85714286
1	0.99491612	0.00365016	0.00143372	2.84261748
2	0.98027167	0.01310457	0.00662376	2.8007762
3	0.93086486	0.04441091	0.02472423	2.65961388
4	0.79307926	0.12612967	0.08079106	2.26594075
5	0.54742509	0.24203971	0.2105352	1.56407169
6	0.31806937	0.28136078	0.40056985	0.90876963
7	0.18928542	0.22849002	0.58222456	0.54081548
8	0.12914419	0.15481697	0.71603885	0.36898339
9	0.10079762	0.09642836	0.80277403	0.28799319
10	0.08665821	0.05766775	0.85567404	0.2475949
11	0.07924426	0.0337788	0.88697694	0.22641217
12	0.07522172	0.0195681	0.90521018	0.21491919
13	0.07299226	0.01126724	0.9157405	0.2085493
14	0.07174075	0.00646572	0.92179353	0.20497358
15	0.07103297	0.00370329	0.92526375	0.20295133
16	0.07063094	0.00211879	0.92725026	0.20180269
17	0.07040202	0.0012115	0.92838648	0.20114863
18	0.07027149	0.00069248	0.92903604	0.20077567
19	0.07019699	0.00039573	0.92940728	0.20056282
20	0.07015445	0.00022612	0.92961942	0.20044129
21	0.07013016	0.0001292	0.92974064	0.20037189
22	0.07011628	7.3819E-05	0.9298099	0.20033224
23	0.07010836	4.2175E-05	0.92984947	0.20030959
24	0.07010383	2.4096E-05	0.92987207	0.20029666
25	0.07010124	1.3767E-05	0.92988499	0.20028926
26	0.07009976	7.8652E-06	0.92989237	0.20028504
27	0.07009892	4.4936E-06	0.92989659	0.20028263
28	0.07009844	2.5673E-06	0.929899	0.20028125
29	0.07009816	1.4667E-06	0.92990037	0.20028046
30	0.070098	8.3798E-07	0.92990116	0.20028001

$$\beta=2.5, \gamma=0.7$$

time (days)	S	I	R	R(E)
0	0.999	0.001	0	3.57142857
1	0.99204751	0.00599704	0.00195546	3.54302681
2	0.95229414	0.03429923	0.01340663	3.40105049
3	0.77423604	0.15439812	0.07136584	2.7651287
4	0.40991882	0.3406584	0.24942278	1.46399579
5	0.16899824	0.33347915	0.49752262	0.60356514
6	0.08406682	0.2228932	0.69303998	0.30023865
7	0.05443395	0.13083138	0.81473467	0.19440698
8	0.04244866	0.07318268	0.88436866	0.15160235
9	0.03699029	0.04010188	0.92290783	0.13210817
10	0.03431576	0.02176224	0.943922	0.12255629
11	0.03294951	0.01175257	0.95529791	0.11767684
12	0.03223533	0.006331	0.96143367	0.11512619
13	0.03185729	0.00340595	0.96473675	0.11377605
14	0.03165583	0.00183105	0.96651312	0.11305652
15	0.03154806	0.00098401	0.96746793	0.11267165
16	0.03149031	0.0005287	0.96798099	0.11246539
17	0.03145932	0.00028404	0.96825664	0.11235472
18	0.03144269	0.00015259	0.96840472	0.11229532
19	0.03143376	8.1968E-05	0.96848428	0.11226342
20	0.03142896	4.4032E-05	0.96852701	0.11224628
21	0.03142638	2.3653E-05	0.96854996	0.11223708
22	0.031425	1.2706E-05	0.9685623	0.11223214
23	0.03142426	6.8251E-06	0.96856892	0.11222948
24	0.03142386	3.6662E-06	0.96857248	0.11222806
25	0.03142364	1.9694E-06	0.96857439	0.11222729
26	0.03142353	1.0579E-06	0.96857542	0.11222688
27	0.03142346	5.6827E-07	0.96857597	0.11222666
28	0.03142343	3.0526E-07	0.96857626	0.11222654
29	0.03142341	1.6397E-07	0.96857642	0.11222648
30	0.0314234	8.8082E-08	0.96857651	0.11222644

$$\beta = 3, \gamma = 0.7$$

S	I	R	R(E)
0.999	0.001	0	4.28571429
0.98745479	0.00983293	0.00271228	4.23194911
0.88703635	0.08522759	0.02773606	3.80158435
0.4737379	0.35217197	0.17409013	2.03030529
0.13891631	0.4007443	0.46033939	0.59535562
0.05155478	0.2568196	0.69162562	0.22094907
0.02868568	0.14289772	0.82841661	0.12293861
0.02085045	0.07629434	0.90285521	0.08935905
0.01760899	0.0401104	0.94228061	0.07546708
0.01611713	0.02094609	0.96293678	0.06907343
0.01539015	0.01090353	0.97370632	0.06595779
0.01502508	0.00566693	0.97930799	0.06439318
0.01483884	0.00294295	0.98221821	0.06359504
0.01474306	0.00152771	0.98372923	0.06318454
0.01469359	0.00079288	0.98451353	0.06297251
0.01466798	0.00041146	0.98492056	0.06286276
0.0146547	0.00021351	0.98513178	0.06280588
0.01464782	0.00011079	0.98524139	0.06277638
0.01464425	5.7488E-05	0.98529826	0.06276108
0.0146424	2.983E-05	0.98532777	0.06275315
0.01464144	1.5478E-05	0.98534308	0.06274903
0.01464094	8.0314E-06	0.98535103	0.06274689
0.01464068	4.1674E-06	0.98535515	0.06274578
0.01464055	2.1624E-06	0.98535729	0.06274521
0.01464048	1.122E-06	0.9853584	0.06274491
0.01464044	5.8219E-07	0.98535898	0.06274475
0.01464042	3.0209E-07	0.98535927	0.06274467
0.01464041	1.5675E-07	0.98535943	0.06274463
0.01464041	8.1334E-08	0.98535951	0.06274461
0.01464041	4.2203E-08	0.98535955	0.0627446
0.01464041	2.1898E-08	0.98535957	0.06274459

$\beta=3.5, \gamma=0.7$

time (days)	S	I	R	R(E)
0	0.999	0.001	0	5
1	0.980111907	0.01607049	0.0038176	4.90055954
2	0.754377745	0.18944995	0.05617231	3.77188872
3	0.207754398	0.47816597	0.31407964	1.03877199
4	0.046977273	0.34160456	0.61141817	0.23488637
5	0.018955256	0.18811004	0.7929347	0.09477628
6	0.011666134	0.09832093	0.89001294	0.05833067
7	0.009070993	0.05059439	0.94033462	0.04535496
8	0.007972598	0.02587852	0.96614889	0.03986299
9	0.007463923	0.01320134	0.97933474	0.03731961
10	0.007217223	0.00672586	0.98605692	0.03608611
11	0.007094723	0.00342459	0.98948069	0.03547362
12	0.007033162	0.00174315	0.99122369	0.03516581
13	0.007002034	0.00088714	0.99211083	0.03501017
14	0.006986246	0.00045146	0.9925623	0.03493123
15	0.006978225	0.00022973	0.99279204	0.03489112
16	0.006974147	0.0001169	0.99290895	0.03487073
17	0.006972073	5.9486E-05	0.99296844	0.03486036
18	0.006971018	3.0269E-05	0.99299871	0.03485509
19	0.006970481	1.5403E-05	0.99301412	0.0348524
20	0.006970208	7.8376E-06	0.99302195	0.03485104
21	0.006970069	3.9881E-06	0.99302594	0.03485034
22	0.006969998	2.0294E-06	0.99302797	0.03484999
23	0.006969962	1.0326E-06	0.99302901	0.03484981
24	0.006969944	5.2546E-07	0.99302953	0.03484972
25	0.006969934	2.6738E-07	0.9930298	0.03484967
26	0.006969929	1.3605E-07	0.99302993	0.03484965
27	0.006969927	6.9231E-08	0.99303	0.03484964
28	0.006969926	3.5228E-08	0.99303004	0.03484963
29	0.006969925	1.7926E-08	0.99303006	0.03484963
30	0.006969925	9.1215E-09	0.99303007	0.03484962

$$\beta=4, \gamma=0.7$$

time (days)	S	I	R	R(E)
0	0.999	0.001	0	5.71428571
1	0.9684306	0.02613076	0.00543864	5.53388915
2	0.54828266	0.34672367	0.10499367	3.1330438
3	0.0807944	0.47910735	0.44009825	0.46168229
4	0.01788912	0.27816254	0.70394834	0.10222355
5	0.00788214	0.14474059	0.84737728	0.04504078
6	0.00517166	0.07370521	0.92112313	0.02955235
7	0.00417604	0.03728066	0.95854329	0.02386311
8	0.0037485	0.01880668	0.97744482	0.02142
9	0.00354984	0.00947598	0.98697418	0.0202848
10	0.00345379	0.0047719	0.99177431	0.01973596
11	0.00340642	0.00240237	0.99419121	0.01946526
12	0.00338282	0.00120928	0.9954079	0.0193304
13	0.003371	0.00060868	0.99602032	0.01926287
14	0.00336507	0.00030636	0.99632857	0.01922897
15	0.00336209	0.00015419	0.99648372	0.01921193
16	0.00336059	7.7607E-05	0.99656181	0.01920336
17	0.00335983	3.906E-05	0.99660111	0.01919905
18	0.00335945	1.9659E-05	0.99662089	0.01919688
19	0.00335926	9.8944E-06	0.99663084	0.01919578
20	0.00335917	4.9799E-06	0.99663585	0.01919523
21	0.00335912	2.5064E-06	0.99663838	0.01919496
22	0.00335909	1.2615E-06	0.99663965	0.01919482
23	0.00335908	6.349E-07	0.99664028	0.01919475
24	0.00335907	3.1955E-07	0.99664061	0.01919471
25	0.00335907	1.6083E-07	0.99664077	0.0191947
26	0.00335907	8.0946E-08	0.99664085	0.01919469
27	0.00335907	4.074E-08	0.99664089	0.01919468
28	0.00335907	2.0505E-08	0.99664091	0.01919468
29	0.00335907	1.032E-08	0.99664092	0.01919468
30	0.00335907	5.1941E-09	0.99664093	0.01919468

Now we introduce changing recovery rates.

$$\beta = 3, \gamma = 0.5$$

time	S	I	R	R(E)
0	0.999	0.001	0	6
1	0.98578401	0.01199641	0.002219585	5.91470404
2	0.85088719	0.12236694	0.02674587	5.10532314
3	0.35014725	0.47511925	0.174733502	2.10088348
4	0.07410341	0.49234772	0.433548877	0.44462045
5	0.02131878	0.33748684	0.641194379	0.1279127
6	0.00943835	0.21356603	0.776995624	0.05663009
7	0.00566925	0.13238123	0.861949513	0.03401553
8	0.00413851	0.08145848	0.914403006	0.02483108
9	0.00341099	0.04996358	0.946625431	0.02046593
10	0.00302986	0.03059688	0.966373269	0.01817914
11	0.00281789	0.01872089	0.978461227	0.01690732
12	0.0026956	0.0114489	0.985855497	0.01617362
13	0.00262346	0.00699967	0.990376872	0.01574075
14	0.00258031	0.00427876	0.993140931	0.01548185
15	0.00255428	0.00261526	0.99483046	0.01532571
16	0.00253851	0.00159839	0.995863099	0.01523104
17	0.00252891	0.00097687	0.996494217	0.01517348
18	0.00252307	0.00059701	0.996879924	0.0151384
19	0.0025195	0.00036485	0.997115646	0.01511701
20	0.00251732	0.00022297	0.997259703	0.01510395
21	0.00251599	0.00013626	0.99734774	0.01509597
22	0.00251518	8.3275E-05	0.997401542	0.0150911
23	0.00251469	5.0891E-05	0.997434422	0.01508812
24	0.00251438	3.1101E-05	0.997454516	0.0150863
25	0.0025142	1.9006E-05	0.997466795	0.01508519
26	0.00251409	1.1615E-05	0.9974743	0.01508451
27	0.00251402	7.0983E-06	0.997478886	0.0150841
28	0.00251397	4.338E-06	0.997481688	0.01508384
29	0.00251395	2.651E-06	0.997483401	0.01508369
30	0.00251393	1.6201E-06	0.997484448	0.01508359

$\beta=3, \gamma=1$

time	S	I	R	R(E)
0	0.999	0.001	0	3
1	0.98953009	0.00729504	0.00317487	2.96859027
2	0.9258477	0.04880396	0.02534834	2.7775431
3	0.65221742	0.20565698	0.1421256	1.95665226
4	0.28752427	0.29732656	0.41514917	0.8625728
5	0.13451784	0.19712951	0.66835266	0.40355351
6	0.08715406	0.09982012	0.81302582	0.26146217
7	0.07064994	0.04634421	0.88300585	0.21194981
8	0.06419006	0.02084113	0.91496882	0.19257017
9	0.06149812	0.00925248	0.9292494	0.18449436
10	0.06034263	0.00408538	0.93557199	0.18102789
11	0.05983995	0.00179964	0.93836041	0.17951986
12	0.05961996	0.00079194	0.9395881	0.17885989
13	0.05952344	0.00034834	0.94012822	0.17857031
14	0.05948103	0.00015319	0.94036578	0.17844309
15	0.05946239	6.7363E-05	0.94047025	0.17838718
16	0.0594542	2.962E-05	0.94051618	0.17836259
17	0.0594506	1.3024E-05	0.94053638	0.17835179
18	0.05944901	5.7269E-06	0.94054526	0.17834703
19	0.05944831	2.5181E-06	0.94054917	0.17834494
20	0.05944801	1.1072E-06	0.94055088	0.17834403
21	0.05944787	4.8685E-07	0.94055164	0.17834362
22	0.05944781	2.1407E-07	0.94055197	0.17834344
23	0.05944779	9.4127E-08	0.94055212	0.17834337
24	0.05944778	4.1388E-08	0.94055218	0.17834333
25	0.05944777	1.8198E-08	0.94055221	0.17834332
26	0.05944777	8.0019E-09	0.94055222	0.17834331
27	0.05944777	3.5184E-09	0.94055223	0.17834331
28	0.05944777	1.5471E-09	0.94055223	0.17834331
29	0.05944777	6.8025E-10	0.94055223	0.17834331
30	0.05944777	2.9911E-10	0.94055223	0.17834331

$\beta=3, \gamma=1.5$

time	S	I	R	R(E)
0	0.999	0.001	0	2
1	0.99210277	0.00443319	0.00346404	1.98420554
2	0.96278062	0.01875478	0.0184646	1.92556124
3	0.85757363	0.0661025	0.07632387	1.71514725
4	0.62686913	0.14012238	0.2330085	1.25373825
5	0.40017395	0.14239832	0.45742773	0.8003479
6	0.2830734	0.08640232	0.63052428	0.5661468
7	0.23506145	0.04148464	0.72345391	0.4701229
8	0.21589812	0.0181278	0.76597408	0.43179624
9	0.20816848	0.00762801	0.78420351	0.41633696
10	0.20502328	0.0031611	0.79181563	0.41004656
11	0.20373786	0.00130184	0.7949603	0.40747572
12	0.2032115	0.00053477	0.79625372	0.40642301
13	0.20299579	0.00021945	0.79678476	0.40599159
14	0.20290736	9.0012E-05	0.79700263	0.40581472
15	0.2028711	3.6914E-05	0.79709198	0.4057422
16	0.20285623	1.5138E-05	0.79712863	0.40571247
17	0.20285014	6.2074E-06	0.79714365	0.40570028
18	0.20284764	2.5454E-06	0.79714982	0.40569528
19	0.20284661	1.0438E-06	0.79715234	0.40569323
20	0.20284619	4.28E-07	0.79715338	0.40569239
21	0.20284602	1.755E-07	0.7971538	0.40569204
22	0.20284595	7.1967E-08	0.79715398	0.4056919
23	0.20284592	2.951E-08	0.79715405	0.40569184
24	0.20284591	1.2101E-08	0.79715408	0.40569182
25	0.2028459	4.962E-09	0.79715409	0.40569181
26	0.2028459	2.0347E-09	0.7971541	0.4056918
27	0.2028459	8.3434E-10	0.7971541	0.4056918
28	0.2028459	3.4213E-10	0.7971541	0.4056918
29	0.2028459	1.4029E-10	0.7971541	0.4056918
30	0.2028459	5.7527E-11	0.7971541	0.4056918

$$\beta = 3, \gamma = 2$$

time	S	I	R	R(E)
0	0.999	0.001	0	1.5
1	0.99388517	0.00269276	0.00342207	1.49082775
2	0.98041701	0.00706514	0.01251785	1.47062551
3	0.9470504	0.01734796	0.03560165	1.4205756
4	0.87536866	0.03655824	0.0880731	1.31305299
5	0.75842157	0.05790151	0.18367693	1.13763235
6	0.62928946	0.06260158	0.30810895	0.9439342
7	0.53158544	0.04782067	0.4205939	0.79737816
8	0.47406845	0.02899618	0.49693537	0.71110267
9	0.44419618	0.01547817	0.54032565	0.66629427
10	0.42950476	0.00774727	0.56274797	0.64425714
11	0.42244558	0.00375832	0.5737961	0.63366836
12	0.41908811	0.00179616	0.57911572	0.62863217
13	0.41749861	0.00085235	0.58164905	0.62624791
14	0.4167477	0.00040312	0.58284919	0.62512155
15	0.41639331	0.00019035	0.58341634	0.62458996
16	0.41622613	8.9816E-05	0.58368405	0.6243392
17	0.41614729	4.2365E-05	0.58381035	0.62422094
18	0.41611011	1.9979E-05	0.58386991	0.62416516
19	0.41609258	9.4215E-06	0.583898	0.62413887
20	0.41608431	4.4427E-06	0.58391125	0.62412646
21	0.41608041	2.0949E-06	0.58391749	0.62412062
22	0.41607857	9.8781E-07	0.58392044	0.62411786
23	0.41607771	4.6579E-07	0.58392183	0.62411656
24	0.4160773	2.1963E-07	0.58392248	0.62411595
25	0.4160771	1.0356E-07	0.58392279	0.62411566
26	0.41607701	4.8834E-08	0.58392294	0.62411552
27	0.41607697	2.3027E-08	0.58392301	0.62411546
28	0.41607695	1.0858E-08	0.58392304	0.62411543
29	0.41607694	5.1198E-09	0.58392305	0.62411541
30	0.41607694	2.4142E-09	0.58392306	0.62411541

$\beta=3, \gamma=2.5$

time	S	I	R	R(E)
0	0.999	0.001	0	1.2
1	0.99513283	0.00163504	0.00323213	1.1941594
2	0.98888866	0.00263381	0.00847752	1.1866664
3	0.97903009	0.00414291	0.016827	1.17483611
4	0.96399979	0.00628046	0.02971975	1.15679975
5	0.94225279	0.00901289	0.04873432	1.13070334
6	0.91302428	0.01198213	0.07499358	1.09562914
7	0.87733034	0.01444367	0.10822599	1.05279641
8	0.83837989	0.01555057	0.14606954	1.00605587
9	0.80059837	0.01490549	0.18449614	0.96071805
10	0.7677563	0.01284169	0.21940201	0.92130756
11	0.74170935	0.01012621	0.24816443	0.89005122
12	0.72245306	0.00746167	0.27008528	0.86694367
13	0.70891715	0.00523609	0.28584676	0.85070058
14	0.69972608	0.00355239	0.29672153	0.8396713
15	0.69362768	0.00235612	0.3040162	0.83235322
16	0.68964207	0.00153955	0.30881838	0.82757048
17	0.68706267	0.00099627	0.31194106	0.8244752
18	0.68540382	0.00064068	0.31395549	0.82248459
19	0.68434131	0.00041036	0.31524833	0.82120957
20	0.6836625	0.00026216	0.31607534	0.820395
21	0.68322954	0.00016721	0.31660325	0.81987545
22	0.68295368	0.00010654	0.31693978	0.81954442
23	0.68277804	6.7835E-05	0.31715413	0.81933365
24	0.68266625	4.3174E-05	0.31729058	0.8191995
25	0.68259512	2.7471E-05	0.31737741	0.81911414
26	0.68254987	1.7476E-05	0.31743265	0.81905984
27	0.68252109	1.1117E-05	0.3174678	0.8190253
28	0.68250278	7.0707E-06	0.31749015	0.81900333
29	0.68249113	4.4972E-06	0.31750437	0.81898936
30	0.68248373	2.8603E-06	0.31751341	0.81898047

$\beta=3, \gamma=3$

time	S	I	R	R(E)
0	0.999	0.001	0	1
1	0.99601642	0.00099254	0.00299103	0.99601642
2	0.99307698	0.00097643	0.00594659	0.99307698
3	0.99020628	0.00095223	0.0088415	0.99020628
4	0.98742651	0.00092079	0.0116527	0.98742651
5	0.98475688	0.00088313	0.01435999	0.98475688
6	0.98221317	0.00084041	0.01694642	0.98221317
7	0.97980754	0.00079385	0.01939862	0.97980754
8	0.97754852	0.00074463	0.02170685	0.97754852
9	0.97544121	0.0006939	0.02386489	0.97544121
10	0.9734875	0.00064271	0.0258698	0.9734875
11	0.9716865	0.00059194	0.02772156	0.9716865
12	0.97003497	0.00054237	0.02942265	0.97003497
13	0.96852778	0.00049461	0.03097761	0.96852778
14	0.96715833	0.00044911	0.03239257	0.96715833
15	0.96591895	0.0004062	0.03367485	0.96591895
16	0.96480133	0.0003661	0.03483257	0.96480133
17	0.96379676	0.0003289	0.03587434	0.96379676
18	0.96289642	0.00029465	0.03680893	0.96289642
19	0.9620916	0.00026329	0.03764512	0.9620916
20	0.96137382	0.00023473	0.03839145	0.96137382
21	0.96073502	0.00020884	0.03905615	0.96073502
22	0.96016753	0.00018547	0.039647	0.96016753
23	0.95966422	0.00016445	0.04017132	0.95966422
24	0.95921849	0.00014561	0.0406359	0.95921849
25	0.95882425	0.00012877	0.04104699	0.95882425
26	0.95847594	0.00011374	0.04141031	0.95847594
27	0.95816853	0.00010037	0.04173109	0.95816853
28	0.95789746	8.8498E-05	0.04201404	0.95789746
29	0.95765861	7.7969E-05	0.04226343	0.95765861
30	0.95744829	6.8646E-05	0.04248306	0.95744829

$$\beta = 3, \gamma = 3.5$$

time	S	I	R	R(E)
0	0.999	0.001	0	0.85714286
1	0.9966503	0.00060241	0.0027473	0.85427168
2	0.99524138	0.0003609	0.00439772	0.85306404
3	0.99439966	0.0002155	0.00538484	0.85234257
4	0.99389789	0.00012842	0.00597369	0.85191248
5	0.99359916	7.6442E-05	0.0063244	0.85165642
6	0.99342145	4.5469E-05	0.00653308	0.8515041
7	0.99331578	2.7035E-05	0.00665718	0.85141353
8	0.99325297	1.607E-05	0.00673096	0.85135969
9	0.99321563	9.5513E-06	0.00677481	0.85132769
10	0.99319345	5.6762E-06	0.00680088	0.85130867
11	0.99318026	3.3731E-06	0.00681637	0.85129737
12	0.99317243	2.0045E-06	0.00682557	0.85129065
13	0.99316777	1.1911E-06	0.00683104	0.85128666
14	0.993165	7.0778E-07	0.00683429	0.85128429
15	0.99316336	4.2058E-07	0.00683622	0.85128288
16	0.99316238	2.4991E-07	0.00683737	0.85128204
17	0.9931618	1.485E-07	0.00683805	0.85128154
18	0.99316146	8.8242E-08	0.00683845	0.85128125
19	0.99316125	5.2435E-08	0.0068387	0.85128107
20	0.99316113	3.1158E-08	0.00683884	0.85128097
21	0.99316106	1.8514E-08	0.00683892	0.85128091
22	0.99316101	1.1001E-08	0.00683897	0.85128087
23	0.99316099	6.5372E-09	0.006839	0.85128085
24	0.99316097	3.8845E-09	0.00683902	0.85128083
25	0.99316097	2.3082E-09	0.00683903	0.85128083
26	0.99316096	1.3716E-09	0.00683904	0.85128082
27	0.99316096	8.15E-10	0.00683904	0.85128082
28	0.99316095	4.8429E-10	0.00683904	0.85128082
29	0.99316095	2.8777E-10	0.00683905	0.85128082
30	0.99316095	1.71E-10	0.00683905	0.85128082

$$\beta = 3, \gamma = 4$$

time	S	I	R	R(E)
0	0.999	0.001	0	0.75
1	0.99711142	0.00036557	0.00252301	0.74783357
2	0.99642314	0.00013316	0.00344369	0.74731736
3	0.9961727	4.8444E-05	0.00377885	0.74712953
4	0.99608163	1.7615E-05	0.00390075	0.74706122
5	0.99604852	6.4043E-06	0.00394507	0.74703639
6	0.99603649	2.3282E-06	0.00396119	0.74702736
7	0.99603211	8.4636E-07	0.00396704	0.74702408
8	0.99603052	3.0767E-07	0.00396917	0.74702289
9	0.99602994	1.1185E-07	0.00396995	0.74702246
10	0.99602973	4.0659E-08	0.00397023	0.7470223
11	0.99602965	1.478E-08	0.00397033	0.74702224
12	0.99602963	5.373E-09	0.00397037	0.74702222
13	0.99602962	1.9532E-09	0.00397038	0.74702221
14	0.99602961	7.1005E-10	0.00397039	0.74702221
15	0.99602961	2.5812E-10	0.00397039	0.74702221
16	0.99602961	9.3832E-11	0.00397039	0.74702221
17	0.99602961	3.411E-11	0.00397039	0.74702221
18	0.99602961	1.24E-11	0.00397039	0.74702221
19	0.99602961	4.5076E-12	0.00397039	0.74702221
20	0.99602961	1.6386E-12	0.00397039	0.74702221
21	0.99602961	5.9568E-13	0.00397039	0.74702221
22	0.99602961	2.1654E-13	0.00397039	0.74702221
23	0.99602961	7.8719E-14	0.00397039	0.74702221
24	0.99602961	2.8616E-14	0.00397039	0.74702221
25	0.99602961	1.0403E-14	0.00397039	0.74702221
26	0.99602961	3.7816E-15	0.00397039	0.74702221
27	0.99602961	1.3747E-15	0.00397039	0.74702221
28	0.99602961	4.9974E-16	0.00397039	0.74702221
29	0.99602961	1.8167E-16	0.00397039	0.74702221
30	0.99602961	6.604E-17	0.00397039	0.74702221

Appendix II

Excel Solutions and Error Data

The following tables contain the ode23 and Euler Excel solutions for $h=1$ and $h=0.5$, including error estimates and quotients.

h=1	second order sol				third order sol				
t	S	I	R	N	t	S	I	R	N
0	0.99	0.01	0	1	0	0.99	0.01	0	1
1	0.986886156	0.01101364	0.002100207	1	1	0.986886662	0.01101351	0.002099825	1
2	0.983470248	0.012118	0.004411747	1	2	0.983470777	0.01211788	0.004411341	1
3	0.979727478	0.01331881	0.006953712	1	3	0.979728025	0.01331869	0.006953282	1
4	0.975632651	0.01462142	0.009745929	1	4	0.975633213	0.01462131	0.009745478	1
5	0.971159908	0.01603082	0.012809268	1	5	0.971160478	0.01603073	0.012808797	1
6	0.966282988	0.01755147	0.016165544	1	6	0.966283559	0.01755138	0.016165056	1
7	0.960975573	0.01918704	0.019837389	1	7	0.960976136	0.01918698	0.019836887	1
8	0.955211687	0.02094024	0.023848077	1	8	0.955212232	0.0209402	0.023847568	1
9	0.948966178	0.02281252	0.028221305	1	9	0.948966693	0.02281251	0.028220793	1
10	0.942215271	0.02480382	0.032980911	1	10	0.942215742	0.02480385	0.032980405	1
11	0.934937187	0.02691227	0.038150547	1	11	0.9349376	0.02691235	0.038150055	1
12	0.927111283	0.02913389	0.043753284	1	12	0.927113168	0.02913402	0.043752815	1
13	0.918726514	0.03146232	0.049811164	1	13	0.918726762	0.03146251	0.04981073	1
14	0.909766727	0.03388857	0.056344698	1	14	0.909766868	0.03388882	0.05634431	1
15	0.900226895	0.03640079	0.063372315	1	15	0.900226912	0.0364011	0.063371985	1
16	0.890106119	0.03898411	0.070909772	1	16	0.890106	0.03898449	0.070909515	1
17	0.879409857	0.04162059	0.078969549	1	17	0.879409589	0.04162103	0.078969376	1
18	0.868150496	0.04428927	0.087560231	1	18	0.86815007	0.04428977	0.087560156	1
19	0.85634779	0.04696629	0.096685918	1	19	0.856347202	0.04696685	0.096685951	1
20	0.84402912	0.04962521	0.106345671	1	20	0.844028368	0.04962581	0.106345822	1
21	0.831229536	0.05223742	0.11653304	1	21	0.831228626	0.05223806	0.116533317	1
22	0.817991571	0.05477274	0.127235684	1	22	0.817990514	0.05477339	0.127236092	1
23	0.804364799	0.05720008	0.138435121	1	23	0.80436361	0.05720073	0.13843566	1
24	0.790405143	0.05948823	0.150106624	1	24	0.790403841	0.05948887	0.150107292	1
25	0.77617395	0.06160676	0.162219289	1	25	0.776172562	0.06160736	0.16222008	1
26	0.761736867	0.06352686	0.174736271	1	26	0.76173542	0.0635274	0.174737176	1
27	0.747162547	0.06522225	0.187615204	1	27	0.747161071	0.06522272	0.187616209	1
28	0.732521263	0.06666995	0.200808783	1	28	0.732519788	0.06667034	0.200809873	1

29	0.71788347	0.06785103	0.214265502	1	29	0.717882027	0.06785132	0.214266657	1
30	0.703318397	0.06875109	0.227930514	1	30	0.703317013	0.06875127	0.227931714	1

h=0.5 second order sol				third order sol			
t	S	I	R	t	S	I	R
0	0.99	0.01	0	0	0.99	0.01	0
0.5	0.98847957	0.01049578	0.00102465	0.5	0.988479631	0.01049576	0.001024604
1	0.986886523	0.01101355	0.002099926	1	0.986886585	0.01101354	0.002099879
1.5	0.985217852	0.01155404	0.00322811	1.5	0.985217916	0.01155402	0.003228061
2	0.983470548	0.01211795	0.004411507	2	0.983470614	0.01211793	0.004411456
2.5	0.981641539	0.01270597	0.005652494	2.5	0.981641606	0.01270595	0.005652442
3	0.9797277	0.01331878	0.006953518	3	0.979727767	0.01331877	0.006953465
3.5	0.977725853	0.01395706	0.008317091	3.5	0.977725921	0.01395704	0.008317036
4	0.975632782	0.01462143	0.009745791	4	0.975632851	0.01462141	0.009745735
4.5	0.973445235	0.01531251	0.011242258	4.5	0.973445305	0.01531249	0.011242201
5	0.971159934	0.01603087	0.012809193	5	0.971160004	0.01603086	0.012809135
5.5	0.968773588	0.01677706	0.014449352	5.5	0.968773658	0.01677705	0.014449292
6	0.966282898	0.01755156	0.016165541	6	0.966282968	0.01755155	0.01616548
6.5	0.963684576	0.01835481	0.017960614	6.5	0.963684646	0.0183548	0.017960553
7	0.960975353	0.01918718	0.019837466	7	0.960975422	0.01918717	0.019837403
7.5	0.958151997	0.02004898	0.021799023	7.5	0.958152065	0.02004897	0.021798961
8	0.955211326	0.02094043	0.023848242	8	0.955211392	0.02094043	0.023848179
8.5	0.952150224	0.02186168	0.025988095	8.5	0.952150289	0.02186168	0.025988032
9	0.948965664	0.02281277	0.028221566	9	0.948965726	0.02281277	0.028221503
9.5	0.945654721	0.02379364	0.030551635	9.5	0.94565478	0.02379365	0.030551573
10	0.942214595	0.02480413	0.032981275	10	0.942214651	0.02480414	0.032981213
10.5	0.93864263	0.02584394	0.035513433	10.5	0.938642683	0.02584395	0.035513371
11	0.934936341	0.02691264	0.03815102	11	0.934936389	0.02691265	0.03815096
11.5	0.931093428	0.02800967	0.0408969	11.5	0.931093472	0.02800969	0.040896841
12	0.927111807	0.02913432	0.043753871	12	0.927111846	0.02913434	0.043753814
12.5	0.922989631	0.03028572	0.046724652	12.5	0.922989664	0.03028574	0.046724597
13	0.918725311	0.03146282	0.049811868	13	0.918725339	0.03146285	0.049811816
13.5	0.914317547	0.03266442	0.053018031	13.5	0.914317568	0.03266445	0.053017981
14	0.909765344	0.03388913	0.056345522	14	0.909765358	0.03388917	0.056345475
14.5	0.905068042	0.03513538	0.059796575	14.5	0.905068048	0.03513542	0.059796532
15	0.900225334	0.03640141	0.063373258	15	0.900225333	0.03640145	0.063373219
15.5	0.895237293	0.03768526	0.067077452	15.5	0.895237283	0.0376853	0.067077417
16	0.890104387	0.03898478	0.070910834	16	0.890104368	0.03898483	0.070910804
16.5	0.884827503	0.04029764	0.074874855	16.5	0.884827475	0.04029769	0.074874831
17	0.879407962	0.04162131	0.078970726	17	0.879407925	0.04162137	0.078970707
17.5	0.873847534	0.04295307	0.083199392	17.5	0.873847486	0.04295313	0.083199379
18	0.86814845	0.04429003	0.08756152	18	0.868148392	0.04429009	0.087561513

18.5	0.862313413	0.04562911	0.092057476	18.5	0.862313346	0.04562918	0.092057476
19	0.856345606	0.04696708	0.096687313	19	0.856345529	0.04696715	0.096687319
19.5	0.850248692	0.04830056	0.101450751	19.5	0.850248604	0.04830063	0.101450765
20	0.844026814	0.04962602	0.106347166	20	0.844026717	0.0496261	0.106347188
20.5	0.837684595	0.05093983	0.111375576	20.5	0.837684488	0.05093991	0.111375605
21	0.831227126	0.05223825	0.116534629	21	0.831227009	0.05223832	0.116534667
21.5	0.824659955	0.05351745	0.121822596	21.5	0.824659829	0.05351753	0.121822641
22	0.817989074	0.05477357	0.127237361	22	0.817988938	0.05477365	0.127237414
22.5	0.811220895	0.05600268	0.13277642	22.5	0.811220752	0.05600277	0.132776482
23	0.804362231	0.05720089	0.138436879	23	0.80436208	0.05720097	0.138436949
23.5	0.797420264	0.05836429	0.14421545	23.5	0.797420106	0.05836437	0.144215528
24	0.79040252	0.05948902	0.150108459	24	0.790402355	0.0594891	0.150108545
24.5	0.783316827	0.06057132	0.156111851	24.5	0.783316657	0.0605714	0.156111945
25	0.776171287	0.06160752	0.162221198	25	0.776171112	0.06160759	0.162221299
25.5	0.768974229	0.06259406	0.16843171	25.5	0.768974051	0.06259413	0.168431819
26	0.761734175	0.06352757	0.174738252	26	0.761733993	0.06352764	0.174738368
26.5	0.754459788	0.06440485	0.18113536	26.5	0.754459605	0.06440491	0.181135481
27	0.747159836	0.06522291	0.187617257	27	0.747159652	0.06522296	0.187617385
27.5	0.739843142	0.06597898	0.194177883	27.5	0.739842958	0.06597903	0.194178015
28	0.732518541	0.06667055	0.200810909	28	0.732518357	0.0666706	0.200811047
28.5	0.725194832	0.06729539	0.207509774	28.5	0.725194651	0.06729543	0.207509916
29	0.717880741	0.06785156	0.214267703	29	0.717880562	0.06785159	0.214267849
29.5	0.710584871	0.06833738	0.221077744	29.5	0.710584696	0.06833741	0.221077892
30	0.703315665	0.06875154	0.227932793	30	0.703315494	0.06875156	0.227932944

h=1	Euler solution			
t	S	I	R	N
0	0.99	0.01	0	1
1	0.98505	0.01295	0.002	1
2	0.9786718	0.0167382	0.00459	1
3	0.9704812	0.02158116	0.00793764	1
4	0.96000914	0.02773698	0.01225387	1
5	0.94669527	0.03550347	0.01780127	1
6	0.92988978	0.04520825	0.02490196	1
7	0.90887044	0.05718595	0.03394361	1
8	0.88288313	0.07173607	0.0453808	1
9	0.85121584	0.08905614	0.05972802	1
10	0.81331285	0.10914791	0.07753924	1
11	0.76892715	0.13170403	0.09936883	1
12	0.71829175	0.15599862	0.12570963	1
13	0.66226549	0.18082516	0.15690936	1
14	0.60238835	0.20453726	0.19307439	1

15	0.54078292	0.22523524	0.23398184	1
16	0.47988124	0.24108987	0.27902889	1
17	0.42203398	0.25071915	0.32724686	1
18	0.36912798	0.25348132	0.37739069	1
19	0.32234446	0.24956858	0.42808696	1
20	0.28212093	0.23987839	0.47800067	1
21	0.24828357	0.22574007	0.52597635	1
22	0.2202598	0.20861583	0.57112437	1
23	0.19728496	0.18986751	0.61284753	1
24	0.17855596	0.17062301	0.65082104	1
25	0.16332308	0.15173128	0.68494564	1
26	0.15093247	0.13377564	0.71529189	1
27	0.14083693	0.11711605	0.74204702	1
28	0.13258979	0.10193998	0.76547023	1
29	0.12583169	0.08831008	0.78585823	1
30	0.12027559	0.07620417	0.80352024	1

h=0.01	Micro timestepping			
t	S	I	R	N
1	0.98427136	0.01340738	0.00232126	1
2	0.97768513	0.01731309	0.00500178	1
3	0.9692415	0.02229567	0.00846283	1
4	0.95846863	0.02861196	0.01291941	1
5	0.94480826	0.03655406	0.01863768	1
6	0.92762119	0.04643685	0.02594196	1
7	0.90620879	0.05857206	0.03521916	1
8	0.87985834	0.07322351	0.04691815	1
9	0.84791992	0.0905399	0.06154018	1
10	0.80991804	0.11046584	0.07961612	1
11	0.76569218	0.13264164	0.10166618	1
12	0.71554367	0.15631644	0.12813989	1
13	0.66034824	0.1803129	0.15933885	1
14	0.6015823	0.20308557	0.19533213	1
15	0.54121972	0.22289802	0.23588226	1
16	0.48149302	0.23810162	0.28040536	1
17	0.42456724	0.24744655	0.32798621	1
18	0.37221985	0.25032436	0.3774558	1
19	0.32562417	0.24685675	0.42751907	1
20	0.28529051	0.23780726	0.47690222	1
21	0.25115294	0.22436393	0.52448313	1
22	0.22274072	0.20788125	0.56937803	1
23	0.19936353	0.18966085	0.61097562	1

24	0.18026193	0.17081155	0.64892652	1
25	0.16470587	0.15218984	0.68310428	1
26	0.15204592	0.1343996	0.71355448	1
27	0.14173129	0.1178249	0.74044381	1
28	0.13330842	0.1026757	0.76401588	1
29	0.12641038	0.08903334	0.78455628	1
30	0.12074311	0.07689007	0.80236682	1

h=0.5	Euler solution			
t	S	I	R	N
0	0.99	0.01	0	1
0.5	0.987525	0.011475	0.001	1
1	0.98469204	0.01316046	0.0021475	1
1.5	0.98145229	0.01508417	0.00346355	1
2	0.97775119	0.01727685	0.00497196	1
2.5	0.97352807	0.01977228	0.00669965	1
3	0.96871586	0.02260727	0.00867688	1
3.5	0.96324085	0.02582154	0.0109376	1
4	0.95702276	0.02945748	0.01351976	1
4.5	0.94997489	0.0335596	0.0164655	1
5	0.9420047	0.03817384	0.01982146	1
5.5	0.93301471	0.04334644	0.02363885	1
6	0.922904	0.04912251	0.02797349	1
6.5	0.91157016	0.0555441	0.03288574	1
7	0.89891207	0.06264778	0.03844015	1
7.5	0.88483336	0.07046171	0.04470493	1
8	0.86924664	0.07900225	0.0517511	1
8.5	0.85207853	0.08827014	0.05965133	1
9	0.83327526	0.0982464	0.06847834	1
9.5	0.81280869	0.10888833	0.07830298	1
10	0.79068234	0.12012585	0.08919181	1
10.5	0.76693699	0.13185861	0.1012044	1
11	0.74165518	0.14395456	0.11439026	1
11.5	0.71496402	0.15625026	0.12878572	1
12	0.68703569	0.16855356	0.14441074	1
12.5	0.65808511	0.18064879	0.1612661	1
13	0.62836455	0.19230448	0.17933098	1
13.5	0.59815522	0.20328336	0.19856142	1
14	0.56775647	0.21335377	0.21888976	1
14.5	0.53747322	0.22230164	0.24022514	1

15	0.50760292	0.22994177	0.2624553	1
15.5	0.47842315	0.23612737	0.28544948	1
16	0.45018095	0.24075684	0.30906222	1
16.5	0.42308491	0.24377719	0.3331379	1
17	0.3973003	0.24518408	0.35751562	1
17.5	0.37294737	0.2450186	0.38203403	1
18	0.35010261	0.2433615	0.40653589	1
18.5	0.32880224	0.24032573	0.43087204	1
19	0.30904733	0.23604806	0.45490461	1
19.5	0.29080982	0.23068076	0.47850942	1
20	0.27403876	0.22438374	0.50157749	1
20.5	0.2586663	0.21731783	0.52401587	1
21	0.2446131	0.20963925	0.54574765	1
21.5	0.23179298	0.20149545	0.56671158	1
22	0.22011667	0.19302221	0.58686112	1
22.5	0.20949482	0.18434184	0.60616334	1
23	0.19984015	0.17556232	0.62459753	1
23.5	0.19106905	0.16677719	0.64215376	1
24	0.18310256	0.15806596	0.65883148	1
24.5	0.17586699	0.14949494	0.67463807	1
25	0.16929418	0.14111825	0.68958757	1
25.5	0.16332156	0.13297905	0.70369939	1
26	0.15789197	0.12511073	0.7169973	1
26.5	0.15295348	0.11753815	0.72950837	1
27	0.14845901	0.1102788	0.74126218	1
27.5	0.14436604	0.10334389	0.75229007	1
28	0.1406362	0.09673934	0.76262445	1
28.5	0.13723494	0.09046667	0.77229839	1
29	0.13413114	0.0845238	0.78134506	1
29.5	0.13129682	0.07890574	0.78979744	1
30	0.12870681	0.07360518	0.79768801	1

h=0.01	Micro timestepping			
t	S	I	R	N
0.5	0.98734116	0.01158315	0.0010757	1
1	0.98448318	0.01328286	0.00223396	1
1.5	0.98121551	0.01522231	0.00356218	1
2	0.97748338	0.0174323	0.00508433	1
2.5	0.97322599	0.01994657	0.00682744	1
3	0.96837616	0.02280188	0.00882196	1
3.5	0.96286023	0.02603781	0.01110196	1

4	0.95659803	0.02969648	0.0137055	1
4.5	0.94950315	0.03382201	0.01667484	1
5	0.94148357	0.03845979	0.02005665	1
5.5	0.93244257	0.0436553	0.02390213	1
6	0.92228029	0.04945268	0.02826703	1
6.5	0.91089575	0.05589274	0.03321151	1
7	0.89818962	0.06301057	0.03879981	1
7.5	0.88406771	0.07083262	0.04509967	1
8	0.86844518	0.07937334	0.05218148	1
8.5	0.85125151	0.08863143	0.06011706	1
9	0.83243593	0.09858595	0.06897813	1
9.5	0.81197329	0.10919242	0.07883429	1
10	0.78986985	0.12037942	0.08975073	1
10.5	0.76616851	0.13204598	0.10178551	1
11	0.74095306	0.14406035	0.1149866	1
11.5	0.71435065	0.15626053	0.12938882	1
12	0.6865321	0.16845708	0.14501082	1
12.5	0.65770941	0.18043824	0.16185234	1
13	0.62813049	0.19197756	0.17989196	1
13.5	0.59807102	0.20284338	0.1990856	1
14	0.56782406	0.21280988	0.21936606	1
14.5	0.53768805	0.22166843	0.24064352	1
15	0.50795437	0.22923833	0.2628073	1
15.5	0.47889547	0.23537592	0.28572861	1
16	0.45075465	0.2399811	0.30926425	1
16.5	0.42373828	0.24300071	0.33326101	1
17	0.39801099	0.24442872	0.35756029	1
17.5	0.3736938	0.24430332	0.38200289	1
18	0.35086504	0.24270156	0.4064334	1
18.5	0.32956355	0.23973235	0.4307041	1
19	0.30979342	0.23552841	0.45467817	1
19.5	0.29152967	0.23023828	0.47823205	1
20	0.2747243	0.22401865	0.50125705	1
20.5	0.25931211	0.21702773	0.52366016	1
21	0.24521601	0.20941979	0.5453642	1
21.5	0.23235164	0.20134094	0.56630742	1
22	0.22063118	0.19292611	0.58644271	1
22.5	0.20996631	0.18429724	0.60573645	1
23	0.20027051	0.17556227	0.62416722	1
23.5	0.19146064	0.16681495	0.64172441	1
24	0.18345801	0.15813522	0.65840677	1
24.5	0.17618906	0.14958987	0.67422107	1

25	0.16958564	0.14123361	0.68918075	1
25.5	0.16358509	0.13311018	0.70330473	1
26	0.15813013	0.12525357	0.7166163	1
26.5	0.15316868	0.11768919	0.72914213	1
27	0.14865348	0.11043506	0.74091147	1
27.5	0.14454182	0.10350284	0.75195534	1
28	0.14079516	0.0968989	0.76230594	1
28.5	0.13737877	0.09062513	0.7719961	1
29	0.13426136	0.08467978	0.78105886	1
29.5	0.13141482	0.07905814	0.78952704	1
30	0.1288138	0.07375317	0.79743303	1

Now the error estimates are presented.

Local error estimate for second order solution $h=1$				
t	S	I	R	
0	0	0	0	0
1	5.06443E-07	1.2486E-07	3.81583E-07	
2	5.28751E-07	1.23034E-07	4.05717E-07	
3	5.476E-07	1.18499E-07	4.29101E-07	
4	5.61835E-07	1.10712E-07	4.51123E-07	
5	5.70159E-07	9.91009E-08	4.71059E-07	
6	5.71149E-07	8.30823E-08	4.88067E-07	
7	5.63285E-07	6.20913E-08	5.01194E-07	
8	5.44993E-07	3.56155E-08	5.09377E-07	
9	5.14708E-07	3.24108E-09	5.11466E-07	
10	4.7095E-07	3.52948E-08	5.06245E-07	
11	4.12428E-07	8.00436E-08	4.92472E-07	
12	3.38146E-07	1.30783E-07	4.68929E-07	
13	2.47534E-07	1.86953E-07	4.34488E-07	
14	1.40572E-07	2.47607E-07	3.88179E-07	
15	1.79152E-08	3.11366E-07	3.29281E-07	
16	1.19008E-07	3.76412E-07	2.57405E-07	
17	2.67925E-07	4.40506E-07	1.72581E-07	
18	4.25708E-07	5.01046E-07	7.53381E-08	
19	5.88413E-07	5.55176E-07	3.32373E-08	
20	7.51404E-07	5.99934E-07	1.5147E-07	
21	9.09536E-07	6.3244E-07	2.77096E-07	
22	1.05742E-06	6.50107E-07	4.0731E-07	

23	1.18971E-06	6.50864E-07	5.3885E-07
24	1.30148E-06	6.3335E-07	6.68128E-07
25	1.38848E-06	5.97086E-07	7.91397E-07
26	1.4475E-06	5.42564E-07	9.04941E-07
27	1.47655E-06	4.71272E-07	1.00528E-06
28	1.47496E-06	3.85625E-07	1.08934E-06
29	1.44348E-06	2.88821E-07	1.15466E-06
30	1.38413E-06	1.84624E-07	1.19951E-06

Local error estimate for second order solution h=0.5			
t	S	I	R
0	0	0	0
0.5	6.12949E-08	1.52426E-08	4.60523E-08
1	6.27528E-08	1.51991E-08	4.75537E-08
1.5	6.41294E-08	1.5084E-08	4.90453E-08
2	6.54093E-08	1.48896E-08	5.05197E-08
2.5	6.65759E-08	1.46078E-08	5.19681E-08
3	6.76116E-08	1.42301E-08	5.33815E-08
3.5	6.84976E-08	1.3748E-08	5.47496E-08
4	6.92139E-08	1.31525E-08	5.60614E-08
4.5	6.97397E-08	1.24347E-08	5.7305E-08
5	7.0053E-08	1.15855E-08	5.84675E-08
5.5	7.01311E-08	1.05958E-08	5.95353E-08
6	6.99506E-08	9.45695E-09	6.04937E-08
6.5	6.94875E-08	8.16031E-09	6.13272E-08
7	6.87175E-08	6.69788E-09	6.20196E-08
7.5	6.76161E-08	5.06224E-09	6.25538E-08
8	6.6159E-08	3.24684E-09	6.29121E-08
8.5	6.43223E-08	1.24614E-09	6.30761E-08
9	6.2083E-08	9.44151E-10	6.30271E-08
9.5	5.94191E-08	3.32681E-09	6.27459E-08
10	5.63103E-08	5.90291E-09	6.22133E-08
10.5	5.27383E-08	8.67153E-09	6.14098E-08
11	4.86871E-08	1.16295E-08	6.03167E-08
11.5	4.41439E-08	1.47713E-08	5.89152E-08
12	3.90991E-08	1.80885E-08	5.71875E-08
12.5	3.35471E-08	2.15699E-08	5.5117E-08
13	2.74869E-08	2.52011E-08	5.2688E-08
13.5	2.09223E-08	2.89646E-08	4.98869E-08

14	1.38625E-08	3.28392E-08	4.67017E-08
14.5	6.32233E-09	3.68007E-08	4.3123E-08
15	1.677E-09	4.08209E-08	3.91439E-08
15.5	1.0108E-08	4.48688E-08	3.47607E-08
16	1.89367E-08	4.89096E-08	2.99729E-08
16.5	2.81222E-08	5.29058E-08	2.47836E-08
17	3.76171E-08	5.68172E-08	1.92001E-08
17.5	4.73675E-08	6.06011E-08	1.32336E-08
18	5.73133E-08	6.42131E-08	6.89978E-09
18.5	6.73886E-08	6.76075E-08	2.18958E-10
19	7.75221E-08	7.0738E-08	6.78405E-09
19.5	8.76382E-08	7.35585E-08	1.40798E-08
20	9.76577E-08	7.60236E-08	2.16341E-08
20.5	1.07499E-07	7.80899E-08	2.94087E-08
21	1.17077E-07	7.97164E-08	3.73608E-08
21.5	1.2631E-07	8.08659E-08	4.5444E-08
22	1.35113E-07	8.1505E-08	5.36082E-08
22.5	1.43406E-07	8.16059E-08	6.18005E-08
23	1.51112E-07	8.11464E-08	6.99655E-08
23.5	1.58157E-07	8.01109E-08	7.80461E-08
24	1.64475E-07	7.84909E-08	8.59839E-08
24.5	1.70006E-07	7.62853E-08	9.37205E-08
25	1.74698E-07	7.35008E-08	1.01198E-07
25.5	1.7851E-07	7.01519E-08	1.08358E-07
26	1.81408E-07	6.62608E-08	1.15147E-07
26.5	1.83369E-07	6.18575E-08	1.21512E-07
27	1.84383E-07	5.69788E-08	1.27404E-07
27.5	1.84447E-07	5.16684E-08	1.32779E-07
28	1.83571E-07	4.59756E-08	1.37596E-07
28.5	1.81776E-07	3.99549E-08	1.41821E-07
29	1.7909E-07	3.36649E-08	1.45425E-07
29.5	1.75552E-07	2.71671E-08	1.48385E-07
30	1.7121E-07	2.05254E-08	1.50685E-07

Local error estimate Euler solution h=1			
t	S	I	R
1	0.000778638	0.00045738	0.00032126
2	0.000986668	0.00057489	0.00041178

3	0.001239702	0.00071451	0.00052519
4	0.001540515	0.00087498	0.00066553
5	0.001887003	0.00105059	0.00083641
6	0.002268591	0.0012286	0.00103999
7	0.002661652	0.0013861	0.00127555
8	0.003024782	0.00148744	0.00153735
9	0.003295928	0.00148377	0.00181216
10	0.003394809	0.00131793	0.00207688
11	0.003234965	0.00093761	0.00229736
12	0.002748076	0.00031782	0.00243026
13	0.001917241	0.00051226	0.0024295
14	0.000806056	0.00145169	0.00225775
15	0.000436795	0.00233721	0.00190042
16	0.001611777	0.00298825	0.00137647
17	0.002533251	0.0032726	0.00073935
18	0.003091863	0.00315697	6.5103E-05
19	0.003279717	0.00271183	0.00056789
20	0.003169582	0.00207113	0.00109845
21	0.002869365	0.00137614	0.00149322
22	0.002480921	0.00073458	0.00174634
23	0.002078567	0.00020665	0.00187191
24	0.001705974	0.00018854	0.00189451
25	0.001382793	0.00045856	0.00184135
26	0.001113451	0.00062396	0.00173741
27	0.000894362	0.00070885	0.00160321
28	0.00071863	0.00073573	0.00145436
29	0.000578684	0.00072326	0.00130195
30	0.00046752	0.0006859	0.00115342

Local error estimate Euler solution h=0.5			
t	S	I	R
0.5	0.000183843	0.0001081	7.57E-05
1	0.000208854	0.0001224	8.646E-05

1.5	0.000236782	0.0001381	9.863E-05
2	0.000267812	0.0001554	0.0001124
2.5	0.000302089	0.0001743	0.0001278
3	0.000339695	0.0001946	0.0001451
3.5	0.000380622	0.0002163	0.0001644
4	0.000424736	0.000239	0.0001857
4.5	0.00047174	0.0002624	0.0002093
5	0.000521129	0.0002859	0.0002352
5.5	0.00057214	0.0003089	0.0002633
6	0.000623706	0.0003302	0.0002935
6.5	0.00067441	0.0003486	0.0003258
7	0.000722455	0.0003628	0.0003597
7.5	0.000765655	0.0003709	0.0003947
8	0.00080146	0.0003711	0.0004304
8.5	0.000827024	0.0003613	0.0004657
9	0.000839333	0.0003395	0.0004998
9.5	0.000835396	0.0003041	0.0005313
10	0.000812494	0.0002536	0.0005589
10.5	0.000768486	0.0001874	0.0005811
11	0.000702126	0.0001058	0.0005963
11.5	0.000613368	1.027E-05	0.0006031
12	0.000503592	9.649E-05	0.0006001
12.5	0.000375703	0.0002105	0.0005862
13	0.000234056	0.0003269	0.000561
13.5	8.41947E-05	0.00044	0.0005242
14	6.75928E-05	0.0005439	0.0004763
14.5	0.000214829	0.0006332	0.0004184
15	0.000351446	0.0007034	0.000352
15.5	0.000472326	0.0007515	0.0002791
16	0.000573702	0.0007757	0.000202
16.5	0.00065337	0.0007765	0.0001231
17	0.000710691	0.0007554	4.467E-05
17.5	0.000746425	0.0007153	3.114E-05
18	0.000762431	0.0006599	0.0001025

18.5	0.000761318	0.0005934	0.0001679
19	0.000746093	0.0005197	0.0002264
19.5	0.000719851	0.0004425	0.0002774
20	0.000685541	0.0003651	0.0003204
20.5	0.000645809	0.0002901	0.0003557
21	0.000602906	0.0002195	0.0003835
21.5	0.000558665	0.0001545	0.0004042
22	0.00051451	9.61E-05	0.0004184
22.5	0.000471497	4.461E-05	0.0004269
23	0.000430363	5.776E-08	0.0004303
23.5	0.00039159	3.776E-05	0.0004294
24	0.000355453	6.925E-05	0.0004247
24.5	0.000322072	9.493E-05	0.000417
25	0.000291454	0.0001154	0.0004068
25.5	0.000263526	0.0001311	0.0003947
26	0.00023816	0.0001428	0.000381
26.5	0.000215198	0.000151	0.0003662
27	0.000194465	0.0001563	0.0003507
27.5	0.000175778	0.0001589	0.0003347
28	0.000158956	0.0001596	0.0003185
28.5	0.000143825	0.0001585	0.0003023
29	0.000130222	0.000156	0.0002862
29.5	0.000117992	0.0001524	0.0002704
30	0.000106995	0.000148	0.000255

Error quotients ode23		
S	I	R
8.07043981	8.21492956	8.02425799
8.08373427	8.2630791	8.03087608
8.09920435	8.32734416	8.03838818
8.1173755	8.41755649	8.04695013
8.13897346	8.55387921	8.05675878
8.16503649	8.78531843	8.06806805
8.19711269	9.27029244	8.08121349
8.23762935	10.9692811	8.09665108
8.29063858	3.43279955	8.11502053
8.36347741	5.97921103	8.13725365

8.47098443	6.88279602	8.16476942
8.64845021	7.23017712	8.19984926
9.0055213	7.41846761	8.24642103
10.1404962	7.53997193	8.31188537
10.6828969	7.627604	8.4120596
6.28450102	7.69608399	8.58791252
7.12242457	7.75303439	8.9885341
7.42772305	7.80285575	10.9189109
7.59026538	7.84833678	4.89932701
7.69425934	7.89142032	7.00142426
7.7686834	7.93361998	7.41675969
7.82616386	7.97629105	7.59791294
7.87306525	8.02086028	7.70165176
7.9129322	8.06909677	7.7703766
7.94789011	8.12353807	7.82031524
7.97929244	8.18831208	7.85901288
8.00805153	8.27099664	7.89045465
8.03482334	8.3875976	7.91694905
8.06012097	8.57929751	7.9399352
8.08439327	8.99489484	7.96037011

Error Quotients Euler		
S	I	R
3.965591792	3.968590022	3.962259497
3.91074012	3.917478705	3.902573094
3.863299802	3.873732719	3.848955661
3.825665108	3.843531737	3.801942302
3.801111728	3.832172169	3.763374578
3.794118975	3.844781724	3.735596264
3.81089368	3.896823166	3.721517141
3.860319121	4.009272237	3.726457399
3.955848597	4.234503425	3.753827033
4.120188848	4.726304465	3.810089892
4.399481304	6.395702592	3.902429081
4.92063458	5.953915324	4.039660904
6.288520547	1.906438407	4.235529986
10.62084823	2.950889318	4.513243378
1.542695687	3.497246745	4.933592939

3.081709094	3.913370875	5.722178341
3.714278174	4.27260265	8.813852298
4.098287709	4.59128854	0.974304101
4.351456902	4.872572096	2.880497083
4.510602024	5.129098564	3.674974908
4.595708388	5.400863422	4.040097403
4.623516202	5.862569832	4.245903234
4.609511454	9.252758589	4.367498833
4.567271228	3.523782824	4.436272099
4.507691606	4.36038606	4.47038116
4.438836244	4.556115371	4.479592626
4.366330374	4.613407094	4.472618217
4.293737714	4.619968603	4.453031231
4.223246377	4.599427663	4.424638912
4.155973456	4.566577896	4.390635706

Data Rounding Experiments

Lastly, the data pertaining to the experimentation with rounding is provided, for both Euler and ode23.

15-digit precision	Euler		
t	S	I	R
0	0.999	0.001	0
1	0.9985005	0.0012995	0.0002
2	0.997851724	0.001688376	0.0004599
3	0.99700935	0.002193075	0.00079758
4	0.995916092	0.002847718	0.00123619
5	0.994498048	0.003696218	0.00180573
6	0.992660107	0.004794916	0.00254498
7	0.990280246	0.006215793	0.00350396
8	0.987202557	0.008050323	0.00474712
9	0.983228907	0.010413909	0.00635718
10	0.978109279	0.013450755	0.00843997

7-digit precision		Euler			
t	S	I	R		
0	0.999	0.001	0		
1	0.9985005	0.0012995	0.0002		
2	0.9978517	0.0016884	0.0004599		
3	0.9970093	0.0021931	0.0007976		
4	0.995916	0.0028478	0.0012362		
5	0.9944979	0.0036963	0.0018058		
6	0.9926599	0.004795	0.0025451		
7	0.99028	0.0062159	0.0035041		
8	0.9872023	0.0080505	0.0047473		
9	0.9832286	0.0104141	0.0063574		
10	0.9781089	0.013451	0.0084402		
4-digit precision		Euler			
t	S	I	R		
0	0.999	0.001	0		
1	0.9985	0.0013	0.0002		
2	0.9979	0.0017	0.0005		
3	0.9971	0.0022	0.0008		
4	0.996	0.0029	0.0012		
5	0.9946	0.0038	0.0018		
6	0.9927	0.0049	0.0026		
7	0.9903	0.0064	0.0036		
8	0.9871	0.0083	0.0049		
9	0.983	0.0107	0.0066		
10	0.9777	0.0138	0.0087		
3-digit precision		Euler			
t	S	I	R		
0	0.999	0.001	0		
1	0.999	0.001	0		
2	0.999	0.001	0		
3	0.999	0.001	0		
4	0.999	0.001	0		
5	0.999	0.001	0		
6	0.999	0.001	0		
7	0.999	0.001	0		
8	0.999	0.001	0		
9	0.999	0.001	0		
10	0.999	0.001	0		
2-digit precision		Euler			

t	S	I	R
0	0.999	0.001	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0
10	1	0	0

15-digit precision		Ode23			
t	S	I	R		
0	0.99	0.01	0		
1	0.987333482	0.01063579	0.002030728		
2	0.98450616	0.01130397	0.004189875		
3	0.981511046	0.01200507	0.006483886		
4	0.978341215	0.01273948	0.008919301		
5	0.974989859	0.01350742	0.01150272		
6	0.971450358	0.01430888	0.014240766		
7	0.967716345	0.01514361	0.017140041		
8	0.963781787	0.01601114	0.020207077		
9	0.959641062	0.01691066	0.023448283		
10	0.955289049	0.01784107	0.026869885		
7-digit precision		Ode23			
t	S	I	R		
0	0.99	0.01	0		
1	0.9873335	0.0106358	0.0020307		
2	0.9845062	0.011304	0.0041898		
3	0.9815111	0.0120051	0.0064838		
4	0.9783413	0.0127395	0.0089192		
5	0.9749899	0.0135074	0.0115026		
6	0.9714504	0.0143089	0.0142406		
7	0.9677164	0.0151436	0.0171399		
8	0.9637818	0.0160111	0.0202069		
9	0.9596411	0.0169106	0.0234481		
10	0.9552891	0.017841	0.0268697		
3-digit precision		Ode23			
t	S	I	R		
0	0.99	0.01	0		
1	0.987	0.011	0.002		
2	0.984	0.012	0.004		
3	0.981	0.013	0.006		
4	0.978	0.014	0.009		
5	0.974	0.015	0.012		
6	0.97	0.016	0.015		
7	0.966	0.017	0.018		
8	0.962	0.018	0.021		
9	0.957	0.019	0.025		
10	0.952	0.02	0.029		

2-digit precision	Ode23		
t	S	I	R
0	0.99	0.01	0
1	0.99	0.01	0
2	0.99	0.01	0
3	0.99	0.01	0
4	0.99	0.01	0
5	0.99	0.01	0
6	0.99	0.01	0
7	0.99	0.01	0
8	0.99	0.01	0
9	0.99	0.01	0
10	0.99	0.01	0