Imperial College London

MEng Final year project

Uncertainty Quantification of Epidemic Phenomena and the Parallel Simulator Tool

Author:
Alina Drăgănescu

Supervisor:
William Knottenbelt

Second Marker:
Thomas Heinis

June 18, 2015

~

Abstract

Technological and industrial advances allows for biological and non-biological epidemics to spread faster than the world has ever seen. The analysis of epidemiological models and uncertainty quantification represents one of the best strategies for the control and management of infectious diseases.

The main contributions of this project are a comparison between simulated and analytically derived measures (mean, variance, skewness) regarding the infected counts of an epidemic and a parallelised tool that can provide the user with a quick visualisations of the particularities of their chosen compartmental model. The report will detail the approach taken in deriving both analytical and simulated measures along with a discussion regarding the implementation of the Parallel Simulator Tool.

This project provides valuable insight regarding the potential of using analytically derived measures to accurately characterise compartmental models. However, the methods described have limitations due to the approximations made while deriving the mathematical formulas of the aforementioned analytical measures. Further work in this area could provide a considerable reduction in the computational costs currently associated with epidemiological analysis. In addition, the Parallel Simulator Tool can be improved to further assist these investigations.

~

Acknowledgements

Firstly, I would like to thank Dr. William Knottenbelt for his guidance and contagious enthusiasm throughout this project. Secondly, I would like to thank former PhD student Anton Stefanek, Prof Luca Bortolussi, Dr. Jeremy Bradley and my second marker Thomas Heinis, for their advice and assistance.

Much appreciation to my cariad, Tom, whose moral support was unmeasurable. Furthermore, I would like to thank my family and friends for their unconditional love and encouragement throughout my time at Imperial College.

~

Contents

Chapter 1

Introduction

1.1 Motivation

In a world where travelling across the globe is increasingly quicker and more accessible, the potential of infectious diseases to become pandemics is a frightening reality. The most recent outbreak of Middle East Respiratory Syndrome (MERS) has been mainly reported in Saudi Arabia and South Korea but has also been imported by travellers to at least 25 countries worldwide. Figure 1.1 shows the current effect of the epidemic on the population of South Korea. The unexpected dynamics of an epidemic shows that uncertainty quantification should be considered an effective outbreak management.

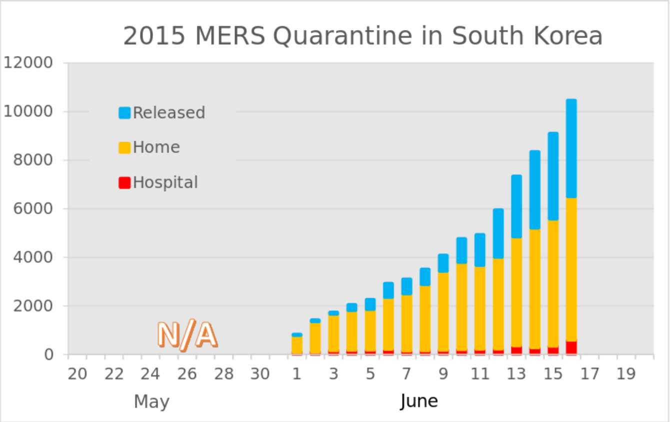


Figure 1.1: MERS quarantine status in South Korea 2015

Epidemic modelling is a research area that looks into analysing, predicting and studying the spread of infectious diseases. Advances in this field can lead to exciting results like predicting the evolution of an epidemic in real time or controlling the way epidemics evolves by using syndemic and counter-syndemic effect to control their course. We call this predictive analytics.

Research in this area can help produce new policies for the control and prevention of epidemic diseases that can save the lives of thousands of people every year. The Global Health Policy Center estimates that 16% of yearly deaths are caused by infectious diseases[1].

Recently, these types of mathematical models have also been used to explain Internet-based phenomenons such as viral videos or songs. In a time where social platforms and other media can potentially reach hundreds of millions people overnight, we are faced with the biggest information pandemics the world has ever seen. For example, if we consider Facebook as an infectious disease, it is the largest epidemic with a recent estimation that 1.44 billion 'infectives' are active monthly[2].

1.2 Objectives

The main objectives of the project are advances into analytical analysis and behaviour of epidemics modelled by compartmental processes. The project is designed to produce both theoretical and practical results. Firstly, we aim to accurately approximate moments of infected counts through an analytical approach and compare the results against synthetic data sets. Secondly, we aim to produce a quick and intuitive tool that can run user customised simulations of compartmental models through a parallelised approach.

These findings would be applicable to both biological and non-biological epidemics, so we are also interested in applying these finding to internet-based phenomenons such as viral videos or business phenomenons such as retail sales.

1.3 Contributions

This project makes the following contributions:

- an analytical derivation of mean, variability, skewness of infected counts in compartmental models
- a comparison between the approximation of moments (mean, variance, skewness) derived analytically and through simulations
- a tool that allows for the visualisation of thousands of stochastic simulations of compartmental models
- a parallelised approach to speed-up the simulation process and produce results in a timely manner

1.4 Report outline

Chapter 2 presents background information regarding epidemic modelling, starting with a short history of epidemics throughout human history and the methods used to combat them. Furthermore, we present the most widely used deterministic mathematical model in the field followed by a stochastic approach to modelling compartmental processes. Next, we present the mathematical foundation of our analytical derivations together with the main sources of uncertainty that compartmental models have to account for. We conclude the chapter by presenting the importance of media analytics and an overview of the tools and libraries used to run experiments.

Chapter 3 presents the details of both analytical and stochastic approaches to studying compartmental processes including the models used, the approximations made by both methods and the result of combining the techniques.

In Chapter 4 we present the server-side architecture of the Parallel Simulator along with design and implementation details of both the frontend and the backend.

Chapter 5 presents the results of our analytical derivations and time speed-ups with a discussion of their interpretation.

Finally, in Chapter 6 we conclude the goals achieved and discuss future work.

~

Chapter 2

Background

2.1 Epidemics throughout history

Human populations have been swept throughout history by waves of diseases and biological epidemics. The theories behind the spread and evolution of infectious diseases have evolved over time as well as the control and prediction of disease outbursts. Hippocrates, the father of medicine, believed that a disease infects the human body because of an imbalance in the four humors (air, fire, water and earth). It was believed that to restore health and cure the sickness, the balance of the humors needs to be restored through practices like bloodletting and dieting. However, there is evidence that, at the time, basic sanitation and prescriptive medicine were also common practices for treating diseases[17].

During the Dark and Middle Ages the world saw a regression of rational theories regarding hygiene and diseases lead by the re-emergence of superstitions and beliefs that diseases are caused by God's wrath. During this time, the world experienced one of the worst pandemics in human history, the bubonic plague, which killed around 25 million people in Europe alone. In an effort to stop the disease, prevention methods such as separating the infected population from the susceptible population came into effect. Later in the 16th century, the theory of small live particles that can spread through water, air or contact was developed. With it, methods were developed to stop the spread of infectious diseases through hygiene. However, until the late 19th century people did not truly understand the nature and spread of bacteria and viruses and hence their efforts to control epidemics were not always successful. Even in recent history, the world has seen deadly epidemics that spread throughout large populations. In Table 2.1 we identify the largest epidemics since the beginning of the 19th century.

Year	Disease
1817 – 1875 AD	Pandemics of cholera
1918	The Spanish Influenza
1940 – now	Lung cancer epidemic
1957	The Asian Influenza
1983 - now	AIDS
1997–now	Obesity pandemic
2003	SARS
2007	Influenza
2014	Ebola

Table 2.1: Major epidemics that killed millions across the world in recent history (19th century - present time).

2.1.1 Traditional control methods

An epidemic is defined as a widespread occurrence of an infectious disease in a community at a particular time. Traditionally, control of epidemic diseases involved a lot of manual effort to collect data on infected patients, their medical history, their interactions, symptoms, treatments they may have received, lab works etc.

A key answer to the evolution of an epidemic is the index case, commonly known as patient zero. This is the first individual that contacted the disease and eventually spread it within a susceptible population. Tracing the medium (air, fluids, direct contact etc.) that facilitated the spread of the disease is another important component of post-epidemic control. Scientists can then create a network of infected individuals linked by the interactions between them that spread the disease, eventually linking all cases to patient zero. Traditional methods used to collect the data are contact tracing and diary-based studies:

- **Contact tracing** - This is the process of identifying and diagnosing the people that came into contact with an infected person. The circle of contact depends on the type of contact required to spread the disease (eg. casual contacts for virulent diseases like Ebola). The methodology is laborious and prone to errors as it relies on individuals to recall day to day interactions.

- **Diary-based studies** - This method assumes that each subject records their interactions in real time. This means that it combats the disadvantages of contact tracing as the work load is shifted onto the subjects rather than onto the scientists. Also, recording interactions as they happen reduces the error rates significantly. However, the methodology introduces other disadvantages. The recordings of individuals might not be consistent or scientists may have difficulties in organising the data due to the fact that patients will make subjective recordings.

2.1.2 Mathematical modelling of epidemics

In order to use mathematical modelling to solve well posed problems, the right model has to be chosen. Factors that influence the choice of model vary from the definition of the problem, the available data, the time available to solve the problem and so on. Initially, research is required to extract the particularities of the problem. In epidemiology, these might be the duration of the infection, the populations susceptible, the length of the incubation period and so on. Next, we choose a model that we consider appropriate for the problem at hand and initialise it with input parameters that are either approximated or derived from data. Finally, we set up the model and perform validation against synthetic and real-life data. If the problem was correctly approximated by the model, we can use it to predict the behaviour of the disease and take appropriate actions.

Modelling has become increasingly more powerful over the last century and is now used in a wide range of fields including epidemiology. Below we mention notable landmarks of mathematical modelling being used in epidemiology.

Bernoulli's Smallpox model Bernoulli's model is the first account of mathematical modelling used to monitor and analyse the spread of an infectious disease. Bernoulli's main interest was to predict the increase in life expectancy if smallpox is ruled out as a cause of death. Using his model, he predicted that the average life expectancy would increase from 26 years and 7 months to 29 years and 9 months[12].

Reed-Frost model This model was developed in the 1920s by Lowell Reed and Wade Hampton Frost and was later published in the 1950s[8]. It is considered to mark the beginning of modern mathematical modelling. It is a simple, iterative deterministic model which predicts how an epidemic will behave over time. Using a set of initial parameters, it can predict how many individuals will be infected and how many will be immune in the next time step.

Kermack-McKendrick model The development of this model was one of the biggest achievements in epidemiology in the 20th century. It assumed that the population can be divided in compartments based on their state (ie. susceptible, infected, recovered etc.) and that individuals are equally likely to transit from one compartment to the other. Mathematical analysis of this model can approximate important information like the duration of an epidemic, the total number of infectives, the maximum number of infective at any particular time point etc.

2.2 Deterministic Compartmental Models

The deterministic model was first defined by Kermack-McKendrick in 1927. In such a model we know with certainty the initial values of the parameters (ie. the number of individuals in each compartment) and the values of the variables in the model (ie. infection rate, recovery rate etc.).

A deterministic model assumes that each state of the system depends on a fixed set of equations and parameters that will be used to decide the transition into the following state. Once the initial conditions have been specified, the system is completely defined. This leads to a deterministic result as multiple runs of the model will yield the same results. This type of model is one of the most used models in epidemiology. When using such a model we make a strong assumption that the population is homogenous[11]. This means that we treat all individuals the same, the only thing differentiating them is the state (compartment) they are in.

In a compartmental model we assume that the population is split into a number of different compartments, each individual belonging to exactly one compartment at any point in time. An individual can change state by moving from one compartment to another.

2.2.1 SIR model

The most common compartmental model is the SIR model, developed in the 1900s by Kermack and McKendrick. In this model, the population is divided into three compartments: susceptible, infected and recovered, defined as followed:

- susceptible - labelled S(t) - This represents the population that is not infected but is susceptible to getting infected if they get in contact with an infectious person.
- infected - labelled I(t) - This represents the set of people that are infected and infectious.
- recovered - labelled R(t) - This represents the individuals that recovered from the disease.

The model assumes a constant population size at all times ie.

$$S(t) + I(t) + R(t) = N \quad \forall t$$

and does not include vital dynamics (births or natural deaths), migration or disease-induced deaths. An individual can change state from susceptible to infected ($S \to I$) or from infected to recovered ($I \to R$). Once an individual has reached the recovered compartment, he/she gains permanent immunity to the disease.

Parameters

- β - the transmission coefficient - This represents the infectiousness of the disease and it determines the number of susceptible individuals that get infected at each time step. More precisely, at a time step, an individual infects $\beta * S(t)$ susceptibles. Hence, the total number of new infectives is $\beta * S(t) * I(t)$.

- γ - the recovery rate - This represents the rate at which infected individuals recover and move into the R compartment. If the average infection duration is $\frac{1}{\gamma}$ units of time, we can make a valid assumption that $\gamma * I$ individuals will recover within a time unit.

The above parameters are used to illustrate the movement of the population using the flow diagram in Figure 2.1.

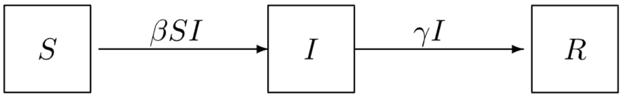


Figure 2.1: Flow diagram of the SIR model. The boxes are the compartments of the model and the values on the arrows represent the population transition rates from one compartment to the other.

We consider that $S(t)$, $I(t)$ and $R(t)$ are differentiable functions of time. Hence, we can now represent the model by the a set of Ordinary Differential Equations (ODEs)[15]:

$$\frac{dS}{dt} = -\beta SI \tag{2.1}$$

$$\frac{dI}{dt} = \beta SI - \gamma I \tag{2.2}$$

$$\frac{dR}{dt} = \gamma I \tag{2.3}$$

The system represents the movement of individuals between the states of the model. These equations coupled with the initial values of the population in each compartment, the transmission coefficient β and the recovery rate γ define the specifics of the epidemic.

The set of initial values in an SIR model must satisfy the following conditions:

$$S(0) = S_0 > 0 \tag{2.4}$$

$$I(0) = I_0 > 0 \tag{2.5}$$

$$R(0) = 0 \tag{2.6}$$

$$S(0) + I(0) + R(0) = S(t) + I(t) + R(t) = N \quad \forall t \tag{2.7}$$

When using differential equations to model disease transmission we assume that the events are occurring continuously. If difference equations are used, then the assumption is that events are taking place at discrete time intervals. Table 2.2 compares the rate of change in the number of individuals in each compartment at time t described by differential equations with the number of individuals in each compartment described by difference equations.

Differential equations - rate	Difference equations - count
$\frac{dS}{dt} = -\beta S(t)I(t)$	$S_{t+1} = S_t - \beta S_t I_t$
$\frac{dI}{dt} = \beta S(t)I(t) - \gamma I(t)$	$I_{t+1} = I_t + \beta S_t I_t - \gamma I_t$
$\frac{dR}{dt} = \gamma I(t)$	$R_{t+1} = R_t + \gamma I_t$

Table 2.2: Comparison of differential equations and difference equations at time t for the SIR model

When modelling an SIR epidemic with difference equations we can encounter accuracy issues in modelling the behaviour of the epidemic. The reason for this problem is that the predicted curve of infected counts becomes less and less smooth with the increase of the time step (e.g. a time step of 2 days). On the other hand, the curve will be closer to the solution of the differential equation as the time step decreases (e.g. a time step of 0.05 days).

An example of an SIR epidemic model with parameters $\beta = 0.001$, $\gamma = 0.1$ and initial population spread $S_0 = 499$, $I_0 = 1$ and $R_0 = 0$ over a time period of 100 days is shown in Figure 2.2.

2.2.2 Epidemic threshold

Transforming the equations 2.1 and 2.2 and letting $\rho = \frac{\gamma}{\beta}$ we obtain the following equation:

$$\frac{dI}{dS} = -1 + \frac{\rho}{S} \tag{2.8}$$

The solutions to the Equation 2.8 in the SI phase plane are shown in Figure 2.3. The curves determined by $I(S)$ reach maximum when $S = \rho$. This

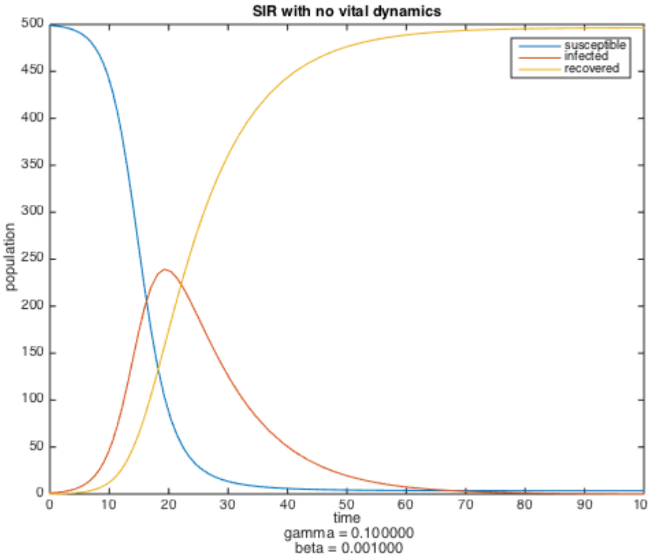


Figure 2.2: Simple SIR model with parameters $\beta = 0.001$, $\gamma = 0.1$ and initial values $S_0 = 499$, $I_0 = 1$, $R_0 = 0$ over a period of 100 days.

shows that there is a threshold for $S(0) = S_0$, the initial number of susceptibles, for which we have the following: if $S_0 > \rho$ the number of infective increases and if $S_0 < \rho$ the number of infective decreases. Define R_0 as:

$$R_0 = \frac{\beta}{\gamma} * S_0 = \frac{S_0}{\rho} \tag{2.9}$$

Then it follows that the epidemic will spread if $R_0 > 1$ and it will die out if $R_0 < 1$. Hence, to control an epidemic a key factor would be the estimation of R_0 and it's reduction to < 1. The ratio $\rho = \frac{\beta}{\gamma}$ can be measured clinically (which is a hard task in practice) and together with an observation of the recovery rate γ, we can determine the transmission coefficient β of the epidemic by $\beta = \frac{\gamma}{\rho}$.

The number R_0 represents the average number of secondary infections produced by one infected individual during the mean course of infection in a completely susceptible population, and is called the basic reproductive number[15].

Measuring the ratio ρ is not always feasible as parameters β or γ might not be known or might not be easily measurable. We can approximate ρ by solving the equation 2.8 with initial value (S_0, I_0):

$$I - I_0 = -S + S_0 + \rho \cdot ln\frac{S}{S_0} \tag{2.10}$$

However, the basic reproductive ratio depends on the disease, the population, the difference in demographic or contact rates, hence estimates of

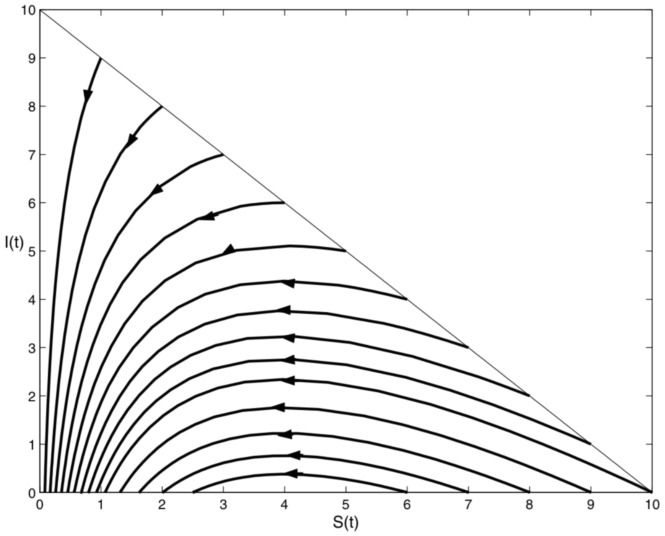


Figure 2.3: Solution orbits of Equation 2.8 for $\beta = 1$ and $\gamma = 4$, reproduced from [15]

Disease	Transmission	R_0
Ebola (2014 outbreak)	Bodily fluids	1.5 - 2.5
Influenza (1918 pandemic strain)	Airborne	2 - 3
SARS	Airborne	2 - 5
HIV/AIDS	Sexual contact	2 - 5
Smallpox	Airborne	5 - 7
Diphtheria	Saliva	6 - 7
Measles	Airborne	12 - 18

Table 2.3: R_0 values for well-known infectious diseases

R_0 for the same disease can yield different values. Table 2.3 presents values of reproductive ratio R_0 of several well-known infectious diseases:

2.2.3 Epidemic burnout

For the SIR model it has been observed that there will always be a certain number of susceptible individuals that do not get infected. This result can be derived mathematically by dividing equation 2.1 by equation 2.3 to obtain:

$$\frac{dS}{dR} = -\frac{\beta S}{\gamma} = -\frac{S}{\rho} = -\frac{R_0}{S_0} \cdot S \tag{2.11}$$

After integration with respect to R we obtain:

$$S(t) = e^{-\frac{R_0}{S_0} \cdot R(t)} \tag{2.12}$$

This shows that the value of the susceptible count is always positive. Hence, we can conclude that the chain of transmission eventually breaks down due to the lack of number of infected, not lack of number of susceptible, which is a counter-intuitive argument[13].

2.2.4 SIR model with vital dynamics

This model preserves the same compartmental split within the population as for the simple SIR but allows for vital dynamics (births and natural deaths). These types of models can have constant or varying population size and can allow for vertical transmission or not. With vertical transmission, the parents could pass the disease to their children at birth. We will look at an SIR model with constant population size and without vertical transmission.

SIR without vertical transmission

In order to model this system we need new parameters that represent the birth rate b and natural death rate μ. The following assumptions are made about the system:

- The population size in our closed environment is constant during the epidemic period, hence the birth and natural death rates are equal ($b = \mu$) and there are no disease-induced deaths.
- There is no vertical transmission, meaning that parents cannot transmit the disease to their children (unlike AIDS for example). Therefore, all newborns enter the susceptible compartment.

We represent the system using an ODE system with initial conditions as follows:

$$\frac{dS}{dt} = bN - \beta SI - \mu S \tag{2.13}$$

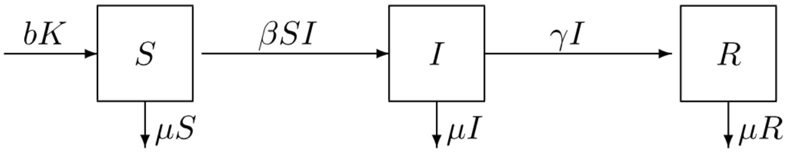
$$\frac{dI}{dt} = \beta SI - \gamma I - \mu I \tag{2.14}$$

$$\frac{dR}{dt} = \gamma I - \mu R \tag{2.15}$$

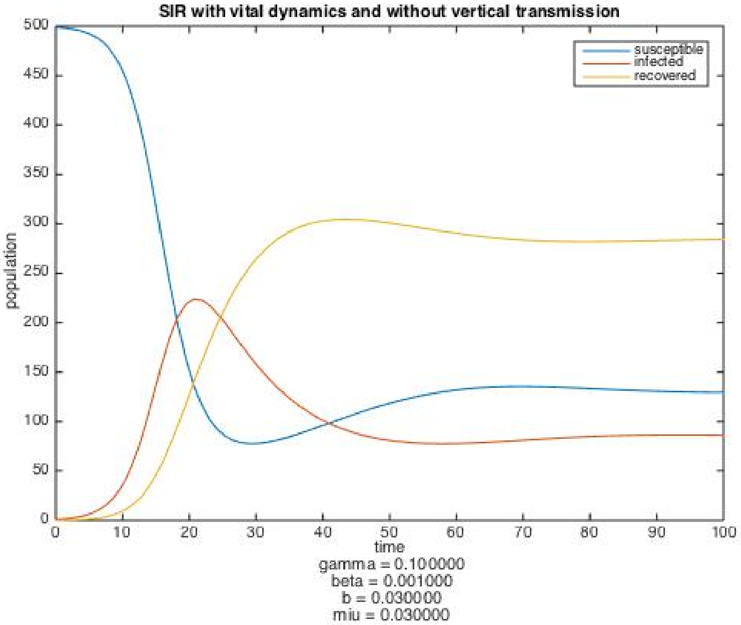
$$S(0) > 0, I(0) > 0, R(0) = 0 \tag{2.16}$$

$$b = \mu \tag{2.17}$$

$$S(0) + I(0) + R(0) = S(t) + I(t) + R(t) = N \quad \forall t \tag{2.18}$$



(a) Flow chart of the SIR model with vital dynamics and no vertical transmission



(b) SIR model with vital dynamics, no vertical transmission, with intial conditions $S_0 = 499$, $I_0 = 1$, $R_0 = 0$ and parameters $\beta = 0.001$, $\gamma = 0.1$ and $b = \mu = 0.03$

Figure 2.4: SIR with vital dynamics flow-chart and plot

2.2.5 SIR model with time-varying parameters

Practice has shown that some types of epidemics are more accurately modelled by time dependent parameters. For example, it has been shown that the spread of measles in the UK during 1948-1966 was driven by school contact and peaked during school terms[21].

We assume that the time-dependent parameters are continuous and bounded functions of time. In our example, we considered the transmission rate as a function of time shown below:

$$\beta(t) = K1 + K2 * sin(\theta t) \tag{2.19}$$

This could, for example, model the spread of flu which peaks during the winter. The equation 2.19 will yield a nonautonomous differential system as the actual time t and starting time t_0 are more important than just the difference between them[3]. This model has proven more difficult to analyse and has been less studied than models without time-variant parameters.

The ODEs representing the system are presented below. Figure 2.5 shows the evolution of this type of epidemic when $K_1 = 0.001$, $K_2 = 0.002$ and $\theta = 2$ in equation 2.19.

$$\frac{dS}{dt} = -\beta(t)SI \tag{2.20}$$

$$\frac{dI}{dt} = \beta(t)SI - \gamma I \tag{2.21}$$

$$\frac{dR}{dt} = \gamma I \tag{2.22}$$

2.2.6 Other types of models

In order to accurately model different behaviours of an epidemic, variations of the SIR model have been developed. The systems used to model the epidemic are chosen to best describe the characteristics of the disease, the environment or the population. We can include the effects of partial immunity, vaccination, migrations etc. However, these models still assume a homogeneous population in which individuals are only differentiated by their state.

SIS model This model is used for diseases in which individuals do not gain immunity after recovering from the disease (e.g. flu). Hence, the infectives become immediately susceptible after recovery. The model is described by the following ODE system:

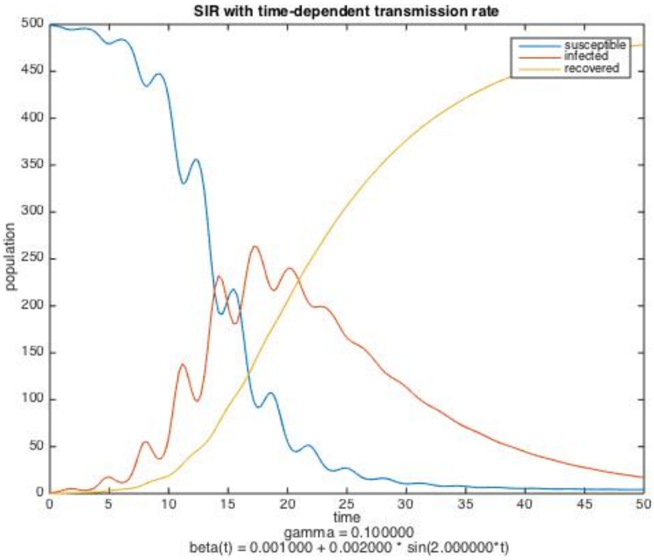


Figure 2.5: SIR model with time-dependent transmission rate $\beta(t) = 0.001 + 0.002 * sin(2t)$, recovery rate $\gamma = 0.1$ and initial values $S_0 = 499$, $I_0 = 1$ and $R_0 = 0$.

$$\frac{dS}{dt} = -\beta SI + \gamma I \tag{2.23}$$

$$\frac{dI}{dt} = \beta SI - \gamma I \tag{2.24}$$

The parameters are the same as in the previous sections and the population size is constant at all times, $S(t) + I(t) = N \quad \forall t$.

SEIR model In this model we introduce a new compartment, E, representing the individuals that were exposed and are infected but are not yet infectious. We model the system with the following set of ODE, where $\frac{1}{\omega}$ is the latent period.

$$\frac{dS}{dt} = -\beta SI \tag{2.25}$$

$$\frac{dE}{dt} = \beta SI - \omega E \tag{2.26}$$

$$\frac{dI}{dt} = \omega E - \gamma I \tag{2.27}$$

$$\frac{dR}{dt} = \gamma I \tag{2.28}$$

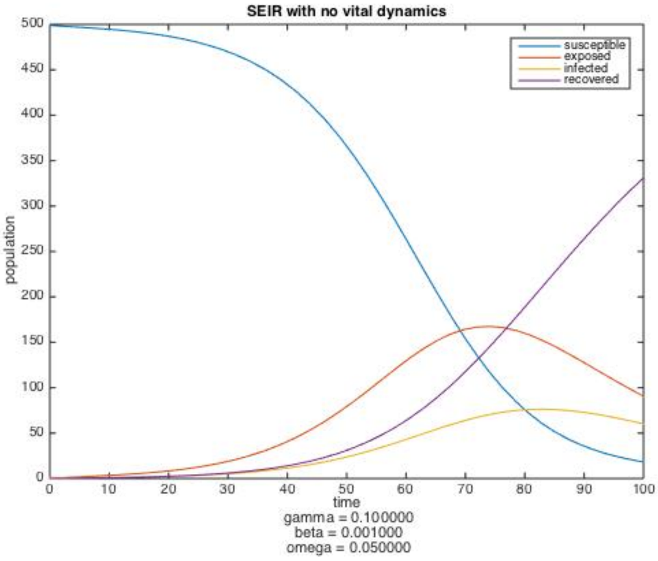


Figure 2.6: SEIR model with no vital dynamics with intial conditions $S_0 = 499$, $I_0 = 1$, $R_0 = 0$ and parameters $\beta = 0.001$, $\gamma = 0.1$ and $\omega = 0.05$

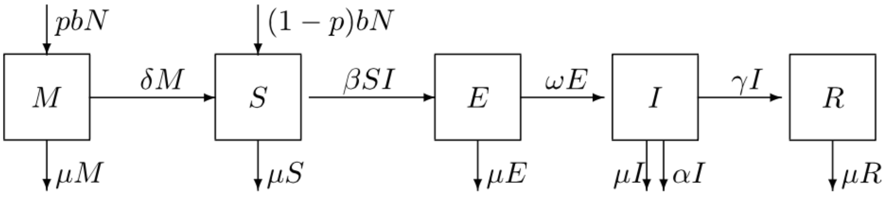


Figure 2.7: Flow chart for MSEIR with vital dynamics

We call ω the progression rate coefficient. We assume that the population is constant, so $S(t) + E(t) + I(t) + R(t) = N \quad \forall t$. The new compartment E introduces a delay in the spread of the epidemic because it takes longer for an individual to move from susceptible (S) to infective (I). Comparing Figure 2.6 with Figure 2.2 we can observe that the peak of the epidemic was delayed by almost 50 time units.

MSEIR model This model incorporates passive immunity for newborns. More precisely, we assume that newborns in this compartment have congenital immunity from maternal antibodies for a few months after they are born, after which they become susceptible. The model does not assume a constant population size as it includes vital dynamics and disease-related deaths. The Figure 2.7 represents the flow chart of the model characterised by the following ODEs:

$$\frac{dM}{dt} = pbN - \delta M - \mu M \tag{2.29}$$

$$\frac{dS}{dt} = (1-p)bN + \delta M - \beta SI - \mu S \tag{2.30}$$

$$\frac{dE}{dt} = \beta SI - \omega E - \mu E \tag{2.31}$$

$$\frac{dI}{dt} = \omega E - \gamma I - \alpha I - \mu I \tag{2.32}$$

$$\frac{dR}{dt} = \gamma I - \mu R \tag{2.33}$$

We assume only a fraction p of the newborns have passive immunity, the rest are born susceptible. The parameter δ represents the fraction of newborns becoming susceptible at each time period, hence the mean period of immunity is $\frac{1}{\delta}$. The rest of the parameters are as follows: b - birth rate coefficient, μ - natural death coefficient, α - disease death coefficient and β, γ, ω as explained above.

2.3 Stochastic Compartmental Models

Deterministic models are useful in deriving certain properties about a system, however they cannot express the randomness needed to model epidemics. More precisely, deterministic models consider individuals to be identical, the only thing differentiating them is their state. This is a very strong assumption which, for example, imposes the rate of infectiousness to be the same for every individual.

Stochastic models attempt to be as close as possible to the actual systems by capturing random elements of the population. This is achieved by associating probabilities with transitions instead of rates (as in deterministic models)[18].

There are different types of stochastic models including Reed-Frost model, discrete time Markov chain, continuous time Markov chain and stochastic differential equations. We are using stochastic models because we are interested in analysing systems that incorporate properties unique to these models, like probability of disease extinction, probability of disease outbreak and expected duration of an epidemic [10]. More precisely, we are using the continuous time Markov chain (CTMC) model in which time is continuous, but the state variable is discrete.

A simulation of an SIR model with no vital dynamics is presented in Figure 2.8. We performed a number of 100 rounds all starting with initial parameters:

$$S_0 = 490, I_0 = 1, R_0 = 0$$
$$\beta = 0.001, \gamma = 0.1$$

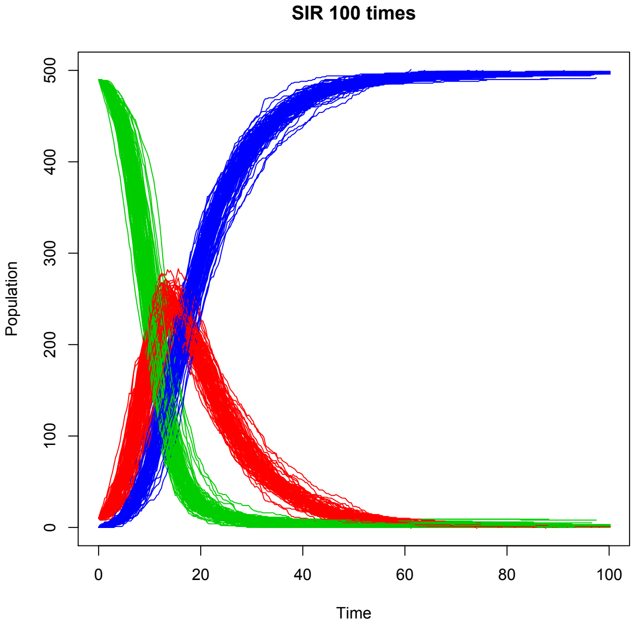


Figure 2.8: 100 stochastic simulation for an SIR model with $S_0 = 490$, $I_0 = 1$, $R_0 = 0$, $\beta = 0.001$, $\gamma = 0.1$. Note that the S count is in green the I count is in red and the R count is in blue.

We can observe slightly different outcomes for different rounds of the stochastic simulation. Analysing this model instead of a deterministic one leads to more accurate results. This is due to the fact that performing a large number of rounds starting from the same parameters and injecting randomness into the model will most likely capture the true state of the epidemic. The generation of these data sets was done using the GillespieSSA package in R (see Section 2.9.1).

Gillespie SSA algorithm The Gillespie Stochastic Simulation Algorithm (SSA) is a procedure for generating statistically correct trajectories of finite well-mixed populations in continuous time[19]. The trajectory that is produced is a stochastic version of the trajectory that would be obtained by solving the corresponding stochastic differential equations. The algorithm has an exact slow version and and three accelerated approximate methods: Explicit tau-leap (ETL), Binomial tau-leap (BTL) and Optimised tau-leap (OTL). We are using the ETL method, see Section 3.3 for more details.

2.4 First and higher order moment calculation from data sets

.

Let $X = x_1, x_2 ... x_n$ be a discrete random variable. We define moments as follows:

- **Mean or expected value**

$$\mu = E[X] = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- **Variance or second central moment**

$$Var[X] = E[X^2] - E[X]^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

  where μ is the mean as defined above. We also note the use of standard deviation, which is the square root of variance.

- **Skewness or third standardised moment**

$$Skew[X] = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^3}{[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2]^{3/2}}$$

  where μ is the mean as defined above.

2.5 Paired t-test and the Null Hypothesis

A paired t-tests is used to compare two population means where we have two samples in which observations in one sample can be paired with observations in the other sample. This might occur in comparing two different methods of measurement. To set up the t-test we have to set the Null Hypothesis and the Alternative Hypothesis and the significance level:

- Null Hypothesis - H_0: $\mu_1 - \mu_2 = 0$
- Alternative Hypothesis - H_a: $\mu_1 - \mu_2 \neq 0$
- Significance Level - 95%

We are omitting the step-by-step details of the test as it can be performed with any statistical application such as R, Matlab, Excel etc. The outcome of these tests is the acceptance or rejection of the Null Hypothesis H_0. The null hypothesis generally states that: “Any differences, discrepancies, or suspiciously outlying results are purely due to random and not systematic errors”. The alternative hypothesis H_a states exactly the opposite. We accept the Null Hypothesis if the p-value reported by the t-test is ≥ 0.05 and it is rejected otherwise.

2.6 ODE analysis of epidemic models

2.6.1 Dynkin’s Formula

Eugene Dynkin[4] is one of the founders of the modern theory of Markov processes. His theorem allows for the calculation of the expected value for any suitable function. It is also seen as stochastic generalization of the (second) fundamental theorem of calculus[5].

Theorem 2.6.1 (Dynkin’s Formula) *Let f be a twice differentiable function with continuous second derivative and $\vec{X} = X_1^{m_1}...X_n^{m_n}$ be a suitably smooth statistic of an Itō diffusion. Then the following formula will hold at time t:*

$$\frac{dE_t[X_1^{m_1}\cdots X_n^{m_n}]}{dt} = \sum_{\tau\in T} E_t[f_\tau(\vec{X})\Big(\prod_{j=1}^{n}(X_j + \boldsymbol{v}_{\tau,j})^{m_j} - X_1^{m_1}\cdots X_n^{m_n}\Big)]$$

For example, using the equation (2.2) of the SIR model, we calculate the second-order moment of the susceptibles to be:

$$\begin{aligned}\frac{dE_t[S^2]}{dt} &= E_t[\beta SI((S-1)^2 - S^2)]\\ &= E_t[\beta SI - 2\beta S^2 I]\\ &= \beta E_t[SI] - 2\beta E_t[S^2 I]\end{aligned}$$

The SIR model is a non-linear quadratic model and hence the equation for a moment of order k depend on moments of order $k+1$. In general, for polynomial rates of maximum degree m, the moments of order k depend on moments of order $k+m-1$.

2.6.2 Moment Closure

In compartmental models, Dynkin’s formula results in a non-linear infinite system of moment approximating ODEs [6]. For example, in compartmental models like the ones presented in Section 2.2, a second-order moment depends on a third-order moment, a third-order moment depends on a fourth-order moment and so on. In order to solve this system numerically it needs to be closed by approximating higher-order moments in terms of lower-order moments. The approximation will transform the infinite system of ODEs into a non-linear closed system of ODEs that can now be solved numerically. This is called moment closure.[22].

To approximate higher-order moments in terms of lower-order moments, moment closure approaches assume that the population at each time point is drawn from a particular family of probabilities. We describe below two types of moment closure: *normal* and *log-normal* which draw their name from the distributions they come from.

Normal Closure Normal moment closure can be applied to any system of ODEs deduced from Dynkin's equation (2.6.1) for which we want to find higher-order moments. For this type of moment closure it is assumed that the populations are approximately multivariate normal at each point in time. It is based on Isserlis' Theorem which allows the computation of higher-order moments of a multivariate normal distribution in terms of its covariance matrix. Hence, higher-order moments from the third onwards can be expressed in terms of mean (first-order) and covariance (second-order)[16].

Theorem 2.6.2 (Isserlis' Theorem) *If $X_1, X_2 ... X_{2n+1}$ are multivariate normal with mean $\vec{\mu}$ and covariance matrix (σ_{ij}) then:*

$$E[(X_1 - \mu_1)(X_2 - \mu_2)...(X_{2n+1} - \mu_{2n+1})] = 0$$

$$\begin{aligned} E[(X_1 - \mu_1)(X_2 - \mu_2)...(X_{2n} - \mu_{2n})] &= \sum \prod E[(X_i - \mu_i)(X_j - \mu_j)] \\ &= \sum \prod COV(X_i, X_j) \end{aligned}$$

where $\sum \prod$ sums through all the distinct partitions of $1...2n$ into disjoint sets of pairs (i, j). If some of the variables appear multiple times then certain pairs will subsequently appear multiple times in the resulting sum.

To obtain the raw moment, we expand the central moment in equation 2.6.2. For example in order to close a system of ODEs at the second-order moment we need the approximation for the joint moment $E[X_1^2(t)X_2(t)]$ as shown below:

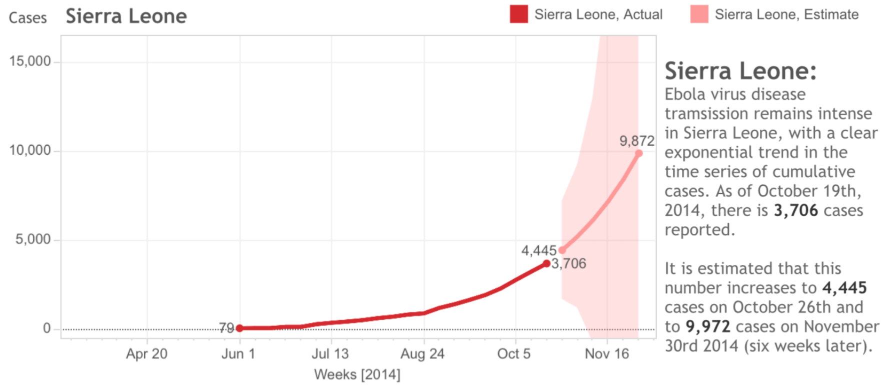
$$\begin{aligned} E[X_1^2(t)X_2(t)] \approx\ & 2E[X_1(t)] \cdot E[X_1(t)X_2(t)] + E[X_1^2(t)] \cdot E[X_2(t)] \\ & - 2E[X_1(t)]^2 \cdot E[X_2(t)] \end{aligned}$$

which yields $E[(X_1 - \mu_1)^2(X_2 - \mu_2)] = 0$ as required since the normal distribution has skewness zero.

2.7 Uncertainty Sources

We model epidemics using discrete or stochastic compartmental models that try to approximate the real world situation as accurately as possible. Through this, we introduce uncertainty sources at every step of the modelling process. We identify three main types of uncertainty explained below:

Measurement Uncertainty When collecting data about an epidemic we perform a variety of measures such as how many individuals became infected during the entire epidemic, how many new infectives per time unit or how many deaths were caused by the infection. These measures are all incomplete and not entirely reliable because we cannot guarantee that we can identify every individual that was affected by the epidemic. Factors like spikes in the number of deaths, general panic among the population, fear of admitting that you are infected, distrust in the medical system, remote populations etc. prevent us from getting an accurate measurement of important data about the epidemic. Figure 2.9 shows the prediction made by The World Health Organisation (WHO) on the number of Ebola infected individual by the end of November 2014. The actual count of infected individuals at that date was around 6.000 cases, far less than the 9.800 predicted. In the context of internet-based phenomenons these measurements are a lot more precise.



Sierra Leone:
Ebola virus disease tramsission remains intense in Sierra Leone, with a clear exponential trend in the time series of cumulative cases. As of October 19th, 2014, there is **3,706** cases reported.

It is estimated that this number increases to **4,445** cases on October 26th and to **9,972** cases on November 30rd 2014 (six weeks later).

Figure 2.9: WHO prediction of Ebola infected population in Sierra Leone in the context of the 2014 Ebola outbreak.

Parameter Uncertainty In all the models presented so far we required a set of parameters to describe certain features of the epidemic (transmission rate, recovery rate etc.). These parameters introduce uncertainty into the system as they are usually based on observation and hence are objective measures of the disease's characteristics. Also, measurements themselves introduce uncertainty by means of completeness and correctness. For example, hospital records could be lost or patients might die before even reaching the hospital. We can estimate the level of uncertainty by looking at the sample sizes or variance of observations for the epidemic.

Stochastic Uncertainty Stochastic models introduce uncertainty by the randomness injected in the evolution of the epidemic. Computationally, stochastic uncertainty can be simulated using Gillespie's discrete-event simulation algorithm (SSA)[18]. A large number of runs of the stochastic algorithm can define confidence levels for the epidemic model but cannot accurately approximate the level of uncertainty introduced by the system.

2.8 Epidemiological models in non-biological phenomenons

With the rise of technology and social media, epidemiological models have been applied to non-biological phenomenons such as viral videos, shared events on social media platforms, computer viruses, business strategies and so on.

We are going to look at how to apply a compartmental model to two non-biological phenomenons:

Viral videos This can be modelled by an SIR process where the population split is as follows:

- Susceptible compartment (S) - every individual that has access to tools that can reproduce the video (eg. YouTube).
- Infected compartment (I) - individuals that have seen the video and decide to share it (ie. they are infected and infectious).
- Recovered compartment (R) - individuals that have seen the video and are no longer sharing it.

Computer viruses Certain computer viruses have a latent period in which they infect the host machine but are not infectious yet. In this case, an SEIR model is more appropriate:

- Susceptible compartment (S) - the set of vulnerable machines that can be accessed directly or through the network.
- Exposed compartment (E) - the machines that are currently infected but are not spreading the virus.

- Infected compartment (I) - the set of infected machines that are now infectious and are trying to spread the disease (as we assume computer viruses are malicious in nature).

- Recovered compartment (R) - the machines that have removed the virus from the system and are no longer infected or infectious.

In both cases, the rates defining the model (ie infection rate, recovery rate) can be approximated through mathematical calculations or from data, similarly to an epidemic modelled by an SIR process.

The advantage of using compartmental models to look at viral phenomenons driven by technology is that data is cheap and widely more available. For example, if we want to trace the popularity of a video on YouTube, we can gather accurate data on the number of views, the number of shares, daily breakdown of views etc. This obviously does not map directly to the number of people that have been 'infected' by the video. However, it is a good approximation, certainly better than traditional methods used to collect data on infectious diseases, mentioned in Section 2.1.1.

2.8.1 Media Analytics

Business analytics are used to make smarter decisions that lead to better business outcomes[9].

A similar approach is applicable to media analytics. The four stages are described below, increasing in value and complexity also represented by the Figure 2.10

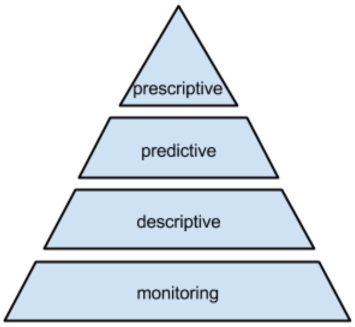


Figure 2.10: Media analytics - correlation between value and complexity

- **Monitoring** - This consists of gathering data and information from events in the past. This is the simplest and cheapest task but it is also the one that gives the least returns in terms of value. We are currently able to gather huge amounts of data and information using very little resources. However, having this information does give any insights except the fact that the data exists.

e.g. We can collect information that can tell us how many units Apple sold within a certain time period, when sales have increased, or that what the evolution of the Apple share price was during the last few years.

- **Descriptive** - This consists of characterising the data available to explain and understand past events. This task answers the question 'What has happened?'. Therefore, it consists mainly of analysing data and fitting it to a certain model. The task is slightly more complex and adds value to the information but is still only looking at past phenomenons. This means we can only reflect at the conclusions of our analysis without being able to change anything.

  e.g. We can fit the data of the spread of a YouTube video views to an SIR epidemic model.

- **Predictive** - This task answers the question 'What could happened?'. It is a considerably harder and a more expensive task that has been extensively researched but not yet mastered. However, in certain situations, we can accurately predict the future based on mathematical models.

  e.g We can predict the peak of popularity of a new album release by approximating its evolution to an multimodal SIR.

- **Prescriptive** - With this task we want to answer the question 'What should happen?'. This is the hardest and most complex task that looks at when certain triggers should be released in order to obtain a certain desired outcome. For example, when is the best time to release a counter-syndemic disease in order to minimise the initial epidemic. There is a high value obtained from being able to solve the problem of when/where to act in order to control the evolution of an epidemic.

  eg. We could figure out when is the best time for Apple to release a new iPhone in order to maximise profits for the new product and the returns from the previous generation products.

2.9 Development Environment

The programming languages most suitable for this project are MATLAB and R as they are both powerful and widely used tools for mathematical modelling and analysis. I decided to use both programs in order to complement their advantages and disadvantages. Also, it was a good opportunity to get a better understanding of both pieces of software and compare their ease of usability and coverage of libraries.

R R is a open-source software supported by a community of over 2 million people [7]. It is built by statisticians which makes it ideal for solving mathematical and statistical problems like the ones we are looking at. It has the disadvantage of being less documented and hence the learning curve is steeper than for MATLAB. However, there are a number of tutorials that can be used as resources along with the support of the active community behind the software. As it was built with extensibility in mind, it has a lot of packages built specifically for certain statistical tasks. This includes a package that can perform the exact type of stochastic simulation we require for our epidemic models. It is believed that R is the most complete piece of statistical software so far. We chose to perform most of our computational tasks using R because it provided us with all the tools needed for computation, plotting and integration with a NodeJS app.

MATLAB The main disadvantage of MATLAB is that it is a commercial application that would be expensive to use under normal circumstances. However, it is very well documented and widely used in the mathematical community. This means that it has a smaller learning curve than R. We chose to use Matlab in certain situations for plotting and solving ODE systems.

Node.js Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js is lightweight and efficient and allows for easy extensibility through npm, a package manager for JavaScript. It has modules integrating with R, the Unix shell, embedded JavaScript templates (EJS) and much more. This allows for easy integration with R over the worker cloud of machines. It has the advantage of automated installation, setup and deployment through npm which makes it ideal for the type of app we are building.

2.9.1 Additional libraries

R GillespieSSA The *GillespieSSA* package provides a simple and intuitive interface to various stochastic simulations for generating simulated trajectories of finite population continuous-time models. Currently, it implements a few Monte Carlo procedures for Gillespie's Stochastic Simulation Algorithm (SSA), including direct and approximate methods [20]. One of the models that is currently included in this package is the Kermack-McKendrick SIR model, which we used to generate thousands of rounds of the SIR process. Similarly, it is easily extendible to be able to run simulations for any type of compartmental model including SEIR, SIRS etc. by simply defining the ODE system that characterises the epidemic system.

R deSolve This package contains the function that solves initial value problems of ordinary differential equations (ODEs), partial differential equations (PDEs), differential algebraic equations (DAEs) and delay differential equations (DeDEs). It uses the function `ode` which is the default integration routine with user specified model parameters, state variables and model equations.

R doParallel The *doParallel* library was built for time-consuming computations or large number of similar tasks that can be run independently[14]. An example of such a problem is running multiple Markov Chain Monte Carlo (MCMC) chains simultaneously. Natively, R will not take advantage of the multi-core underlying architecture and hence the *doParallel* library was built to exploit this resource. In order to use this package, initial setup is required and therefore the communication overhead of setting up the cluster is not justified when dealing with simple problems.

R Rserve *Rserve* acts as a socket server (TCP/IP or local sockets) which allows binary requests to be sent to R. It provides a fast binary transport as no R initialisation is required. The package is persistent as each connection has a separate workspace and working directory. Rserve supports remote connection, user authentication and file transfer between the client and the server, hence it can be used as a remote server for tasks such as generating plot images.

ExpressJS This is a minimal NodeJS web application framework. It includes hooks for commonly used functionality such as session storage, route management and security. It acts as an HTTP server, allowing us to serve our website's static assets as well as an API to process data.

NodeJS R Input Output (RIO) module RIO connects an app to Rserve, a TCP/IP server which allows other programs to use facilities of R. It supports a wide range of R objects, including double, integer, string and arrays of these types, along with raw vectors i.e. images or files. Using this module, we can call R scripts using JSON objects as parameters, perform the desired R functions and serialise the response before returning it to NodeJS.

R rJson This small library is used to perform conversion from R objects to JSON objects and vice-versa. Coupled with the Node RIO module and Rserve library, it facilitated the data transfers between R scripts and the JavaScript backend in our system.

Chapter 3

Analytical and stochastic approximations

We model epidemics using compartmental models defined by a set of ODEs that express the rates at which the populations move from one compartment to the other. Solving the system deterministically is computationally cheap and gives a quick idea of the behaviour and some particularities of the model. However, this approach assumes the rates of transfer between compartments is identical for every individual, which hides vital information that gives the model accuracy. To overcome this issue, stochastic modelling is used to generate thousands of trajectories that behave slightly different in order to capture the true nature of the model. These computations are expensive, especially when we consider that the desired number of simulations lie within 10^5 - 10^7. Hence, we are looking at a way of capturing some features of the stochastic simulations with analytical computations.

We are interested in estimating the moments of the infected count for different compartmental models. In the first section of this chapter, we present the details of the analytical computation of the mean, variance and skewness based on solving ODE systems and moment closures. Next, we show the computations of the same moments using stochastic simulations of the compartmental models. Chapter 5 presents a comparison between the results of the two approaches.

3.1 Analytical calculation of variability

Any compartmental model is characterised by a set of ODEs that define the rates and the movements of the populations from one compartment to the other. The system can be solved analytically with any ODE solver package. Below we have an illustration of the SIR model trajectory for parameters $\beta = 0.001$, $\gamma = 0.1$ and initial populations $S_0 = 250, I_0 = 5, R_0 = 0$ during a period of 100 days in a closed population:

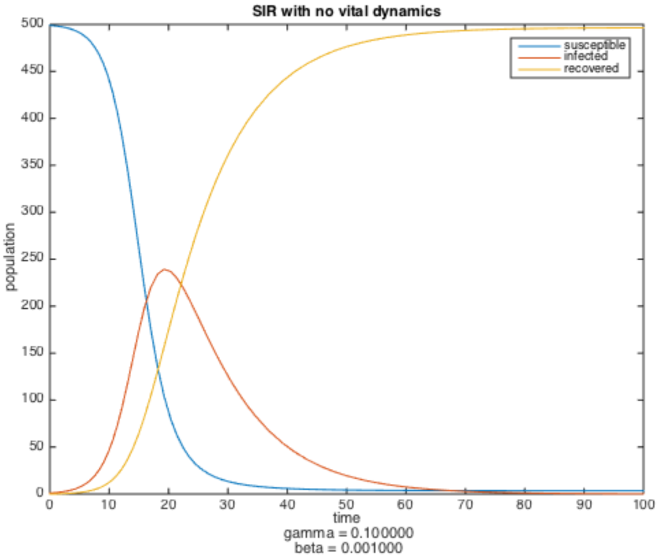


Figure 3.1: Deterministic trajectory prediction for the SIR model with parameters $\beta = 0.001$, $\gamma = 0.1$ and initial conditions $S_0 = 250, I_0 = 5, R_0 = 0$

To compute the variability of the infected count we use the following formula:

$$Var_t(I) = E_t[I^2] - E_t[I]^2 \tag{3.1}$$

This means that an expression for the expected value of I^2 and I respectively is required. We are using Dynkin's formula to obtain expressions of this values.

Applying Dynkin's formula Starting from an SIR model, we can find an expression for the derivative of the expected value of I as shown below: From the initial ODE

$$\frac{dI}{dt} = \beta SI - \gamma I$$

we obtained the following expressions for $dE_t[I]/dt$ and $dE_t[I^2]/dt$ respectively:

$$\frac{dE_t[I]}{dt} = \beta E_t[S]E_t[I] - \gamma E_t[I]$$

$$\begin{aligned}\frac{dE_t[I^2]}{dt} &= E_t[\beta SI((I+1)^2 - I^2) + \gamma I((I-1)^2 - I^2)] \\ &= E_t[\beta SI((I+1)^2 - I^2)] + E_t[\gamma I((I-1)^2 - I^2)] \\ &= 2\beta E_t[SI^2] + \beta E_t[SI] - 2\gamma E_t[I^2] + \gamma E_t[I]\end{aligned}$$

We observe that the expressions depend on the expected values of S, I, R, the expected value of second-order joined moments like $E_t[SI]$ and higher-order moments like $E_t[SI^2]$. More precisely, the equations for a moment of order k depend on moments of order up to $k+1$ due to the fact that the SIR model is quadratic non-linear. In general, if we have polynomial rates of maximum degree m, then moments of order k depend on moments of order $k+m-1$.

This suggests that expressions for first and second order moments for S and R are required. We obtained them in a similar manner using Dynkin's formula:

$$\frac{dE[S]}{dt} = -k_I E_t[I]E_t[S] \tag{3.2}$$

$$\frac{dE[R]}{dt} = k_R E_t[I] \tag{3.3}$$

$$\tag{3.4}$$

$$\frac{dE[S^2]}{dt} = \beta E_t[SI] - 2\beta E_t[S^2 I] \tag{3.5}$$

$$\frac{dE[R^2]}{dt} = 2\gamma E_t[IR] + \gamma E_t[I] \tag{3.6}$$

along with expressions for all second-order joined moment.

In general, for an initial ODE system with n equations we obtain n equations for first-order moments, n equations for second-order moments and $\binom{n}{2} = n(n-1)/2$ equations for second-order joined moments. Therefore, for our SIR model with 3 ODEs we obtain 9 equations. However, these equations depend on third-order moments. If we were to express the third-order moments we would obtain a system which includes moments up to fourth order. Expanding this further will lead to an infinite ODE system that we cannot solve. Hence, to solve the system we need to close it through what it is known as moment closure.

Moment Closure After applying Dynkin's formula, we obtain a system of $n(n+3)/2$ equations for a compartmental model with n equations. For an SIR model, the system of equations depend on third-order moments. We use normal moment closure, explained in Section 2.6.2 to express third-order moments in terms of first and second order moments.

The expression of $E_t[S^2 I]$ in equation 3.5 will be replaced by the approximation:

$$E_t[S^2 I] = 2E_t[S]E_t[SI] + E[S^2]E_t[I] - 2E_t[S]^2 E_t[I] \tag{3.7}$$

The resulting system will now have 9 equations containing moments up to second order and can be solved to find numerical values of $E_t[I]$ and

$E_t[I^2]$ at a time point t. Using equation 3.1 we can derive the numerical expression of the variance of infected count at different time points.

3.2 Analytical calculation of skewness

Skewness is the third standardised moment and is calculated in terms of non-central moments as follows:

$$\begin{aligned}
Skew[X] &= E\left[(\frac{X-\mu}{\sigma})^3\right] \\
&= \frac{E[X^3] - 3\mu E[X^2] + 3\mu^2 E[X] - \mu^3}{\sigma^3} \\
&= \frac{E[X^3] - 3\mu(E[X^2] - \mu E[X]) - \mu^3}{\sigma^3} \\
&= \frac{E[X^3] - 3\mu\sigma^2 - \mu^3}{\sigma^3}
\end{aligned}$$

where μ is the mean of the distribution and σ is the standard deviation.

Similarly to the calculation of variability, we apply Dynkin's formula to obtain an expression for $dE_t[I^3]$. This will increase the size of our ODE system, adding 3 equations for third-order moments for S, I and R respectively. Also, we have to include third-order joined moments for SI^2, SR^2, S^2I, IR^2, RS^2, RI^2 and SIR. We again use normal moment closure to close the system to equations up to and including third-order moments.

3.3 Stochastic simulation of epidemic models

In this section we will define our model used for stochastic simulations followed by details of the data analysis performed on the resulting synthetic data sets.

3.3.1 Model

In order to calculate first, second and third order moments for the synthetic data set, we perform a large number of simulations using the ETL approximated method of the Gillespie algorithm at different time steps. In Figure 3.2 we performed 1,000 runs of an SIR model with initial populations $S_0 = 250$, $I_0 = 5$, $R_0 = 0$ and parameters $\beta = 0.001$ and $\gamma = 0.1$ at a simulation time step of 0.3. As we are interested on the trajectories of infected counts, we omitted the trajectories for susceptible and recovered populations for visibility.

We can observe why stochastic simulations of a compartmental model are more appropriate than a deterministic approach because it does not assume that all individuals are equally likely to change state. Hence, some trajectories simulating the epidemic may finish early having infected a very small population. Performing multiple runs increases the confidence that the true nature of an epidemic will be captured.

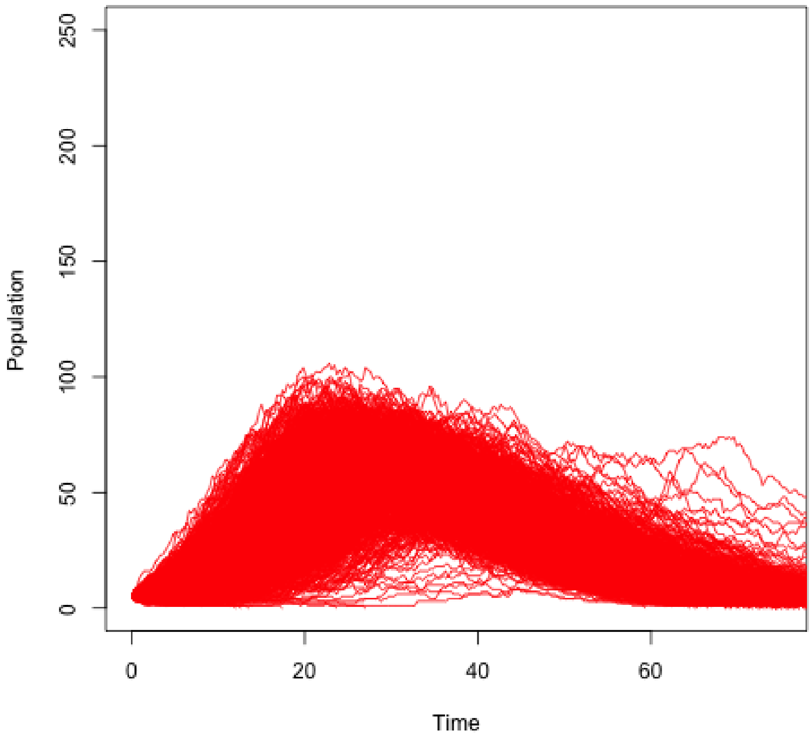


Figure 3.2: 1,000 simulations of an SIR model with parameters $S_0 = 250$, $I_0 = 5$, $R_0 = 0$, $\beta = 0.001$ and $\gamma = 0.1$ at a time step of 0.3 days

Below we present the function used to generate the data sets which we collect for analysis:

```
generateData_SIR <- function(numberOfRuns, susceptible,
                infected, recovered, beta, gamma) {
     sequence <- seq(1,numberOfRuns);
     params <- c(beta=beta, gamma=gamma);
     x0 <- c(S=susceptible, I=infected, R=recovered);
     a  <- c(”beta*{S}*{I}”, ”gamma*{I}”);
     nu <- matrix(c(-1, 0, 1, -1, 0, 1), nrow=3, byrow=T);
     allData <- list();

     for(i in sequence) {
          out <- ssa(x0, a, nu, params, tf=100, method=”ETL”,
                         ignoreNegativeState=TRUE);
          allData[[i]] <- out$data;
     }
     return(allData);
}
```

3.3.2 Data analysis

After the simulation step is finished and all the data is received, we proceed to calculate mean, variance and skewness at each time point t using the formulas presented in Section 2.4.

In addition, we calculate the 95% confidence intervals around the simulated mean using the following formula:

$$\begin{aligned} LowerEndpoint &= \mu - 1.96\frac{\sigma}{\sqrt{n}} \\ UpperEndpoint &= \mu + 1.96\frac{\sigma}{\sqrt{n}} \end{aligned}$$

where μ is the mean value of the distribution at time t, σ is the standard deviation, n is the sample size and 1.96 is the value corresponding to a 95% confidence interval from the z-table.

The interpretation of the confidence interval is that if we repeat the simulation, in 95% of the cases the new mean μ' will be between the endpoints calculated for our current mean μ. Even if the distribution of infected counts at time t is not normally distributed, this formula is valid as we have a large sample data.

Chapter 4

The Parallel Simulator Tool

The parallel variability simulator is a web app tool built to efficiently simulate large data sets and visualisations for compartmental models. The frontend is a simple interface that allows the user to customise the compartmental modal to fit his/her needs. The backend is responsible for parallelising the user's request and for collecting the data from which statistics and graphs are drawn. We are now going to discuss implementation details of both components and justify the reasons for our architectural and implementation choices.

4.1 Backend

In this section we describe the implementation details and decisions made for the backend of the parallel simulator. In Section 4.1.2 we describe the system architecture and illustrate the work-flow using a detailed example of a possible user request.

4.1.1 Implementation decisions

The backend of our parallel simulator is written in JavaScript using the NodeJS web framework Express. Express is a minimal web application framework that provides us with a robust set of features without obscuring NodeJS functionalities. Below we can see the lines of code needed to start a server using Express and NodeJS.

```
var express = require(‘express’);
var app = express();

var server = app.listen(3000, function () {
  var host = server.address().address;
  var port = server.address().port;
  console.log(‘Example app listening at http://%s:%s’,
              host, port);
});
```

NodeJS has numerous self-contained modules that have been developed for the platform. The RIO module described in Section 2.9.1 is an easy to install module that facilitates connections between R and NodeJS. Since our stochastic simulations are performed exclusively in R, the existence of this module made us decide that the NodeJS platform was the right tool for the job. Next, we have an example of a call to an R function from the main server using the RIO module:

```
var rio = require(’rio’);
var statsArgs = { ... }
rio.sourceAndEval(path.join(’./R/calculateStats.R’, {
                  entryPoint: ’calculateStats’,
                  data: statsArgs,
                  callback: function(error, response) {
                      if(!error) {
                          ...
                      } else {
                          ...
                      }
                 }
});
```

Parallel problem and speed-ups Our tool is designed to perform tens of thousands of stochastic simulations and render the result to the user in a friendly and timely mater. However, these simulations are time consuming and computationally heavy and hence would considerably slow down the response time of a user request. In order to increase our performance and due to the fact that the simulations are independent we decided that this problem is perfect for a parallel approach.

We launched a cluster of worker servers on the Imperial College CloudStack. Here we can create and customise a large number of virtual machines that can take the load off our main server. Once the first worker machine was set up with the right configuration to be able to communicate with the main server, we installed all the necessary tools and libraries needed to run the stochastic simulations. Using CloudStack's administrative interface, we were able to transform our worker into a template from which we could create numerous other worker servers which are ready to start processing requests without additional setup. The details of the system architecture are presented in Section 4.1.2.

For an increase speed-up we took advantage of R's own parallel package that can utilise the underlying architecture of the worker machines. For our current cluster of workers, we are using dual core machines. Hence, using this package can offer up to a 50% improvement in the actual simulation time. This is extremely beneficial as it means that we can achieve the a better performance with half the number of machines. A better performance can be achieved due to the overhead of using map-reduce over the network. For example, a single quad-core machine will have better performance than a cluster of 4 single-core worker servers. In Section 5.2 you can see our analysis of the speed improvements achieved using the parallel approach and the R parallel package. Below is an example of the set up required for the cluster of cores used by R's parallel library:

```
cl <- makeCluster(2)
registerDoParallel(cl)

out <- foreach(i=1:numberOfRuns, .inorder=FALSE,
               .packages='GillespieSSA') %dopar% {
    ssa(x0, a, nu, params, tf=100, method="ETL",
        ignoreNegativeState=TRUE);
}

stopCluster(cl)
```

4.1.2 Architecture

The high-level view of the backend architecture is presented in Figure 4.1. Figure 4.2 shows the setup for each individual machine.

We are implementing the master-slave architectural patter. The main server acts as a coordinator and also performs some relatively small computational tasks. The machine cluster is formed by worker servers who only communicate with the coordinator to serve its requests. More precisely, to perform the time-consuming computations. If the user chooses to perform only analytical calculations for a certain model, the main server will perform all the computations and no requests are passed to the cluster of machines.

We are going to explain the details of an end-to-end user request flow involving simulations through the system with the following example:

10,000 simulations of an SIRS model on 8 machines with parameters $\beta = 0.001$, $\gamma = 0.1$, $\delta = 0.02$ and initial conditions $S_0 = 300$, $I_0 = 1$, $R_0 = 0$

1. First, the user request containing all the parameter customisation is received by the main server as a result of the user submitting the filled-in form for the SIRS model.

2. The coordinator calculates the analytical mean, variance and skewness for the given parameters and splits the task of 10,000 simulations (roughly) even between the 8 machines selected by the user. When this is completed, the server now has the complete set of parameters needed by a worker machine to performs its task.

3. Each of the 8 workers receives the HTTP request from the server and starts the simulation. Each machine has an Rserve daemon which processes the R requests in parallel if the R parallel library is enabled. While the workers are busy, the coordinator is waiting for individual replies from the machines. If a machine fails, the coordinator will try to recover by sending the request for the failed machine to a free worker, if one exists, or waits until one becomes available.

4. When the coordinator receives all the responses from the workers it proceeds by processing the data received using its own copy of a parallelised R program. This includes calculating statistics such as mean, variance and skewness at each point time and plotting the trajectories from the simulation. The data is now returned to the user to save, review or analyse further.

If the user chooses to perform an analytic only calculation, then the coordinator will not perform part of step 2, more precisely splitting the number of simulations chosen, and step 3 in its entirety. Also, step 4 will proceed without the server waiting for worker responses.

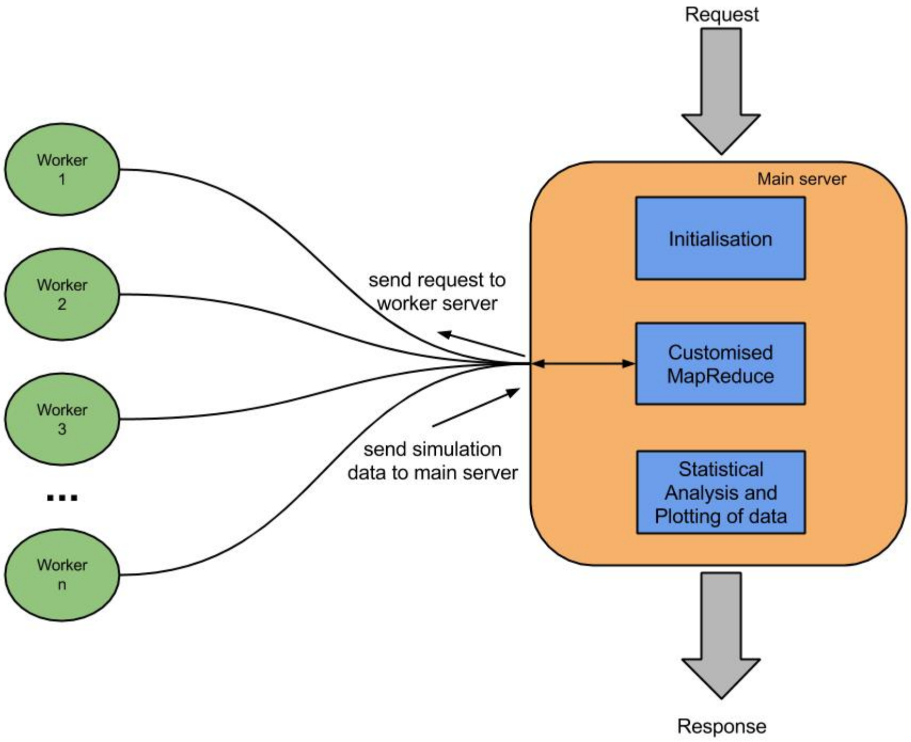

Request
Main server
Initialisation
Customised MapReduce
Statistical Analysis and Plotting of data
Worker 1
Worker 2
Worker 3
...
Worker n
send request to worker server
send simulation data to main server
Response

Figure 4.1: Backend architecture overview

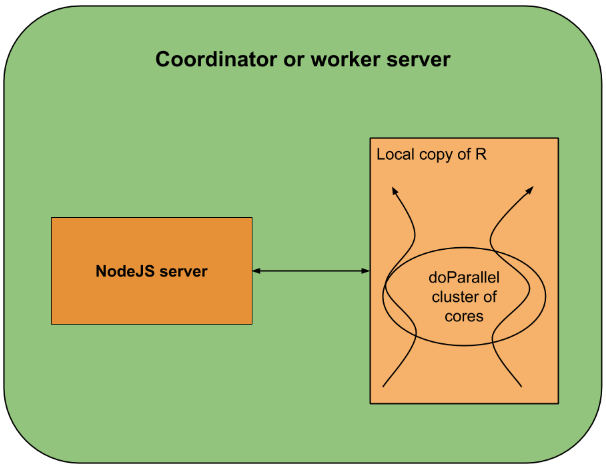
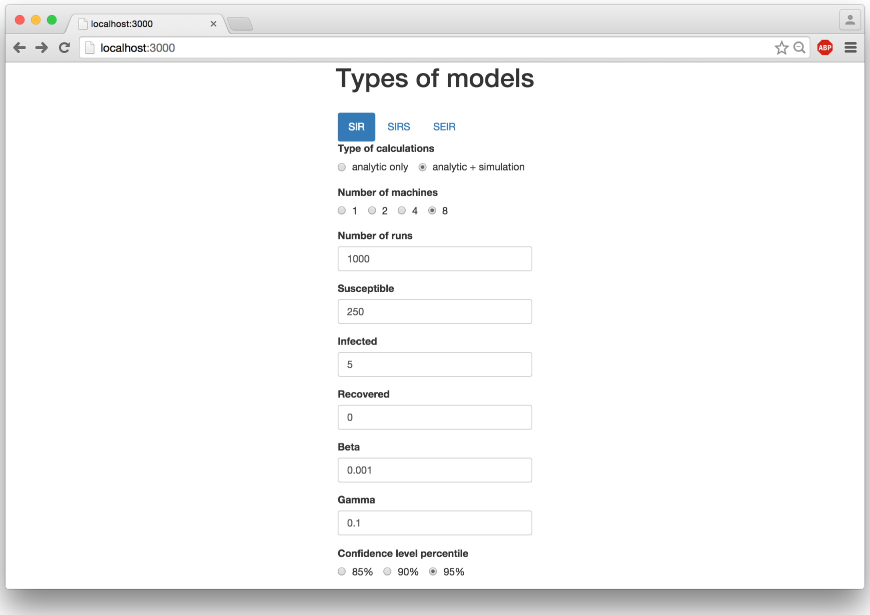


Figure 4.2: Local architecture overview

4.2 Frontend

The frontend was built using the EJS Javascript templating system and Bootstrap. The user is prompted with a simple form that allows him to choose what type of model he would like to generate data for, see Figure 4.3.



Types of models

SIR SIRS SEIR

Type of calculations

○ analytic only ● analytic + simulation

Number of machines

○ 1 ○ 2 ○ 4 ● 8

Number of runs

1000

Susceptible

250

Infected

5

Recovered

0

Beta

0.001

Gamma

0.1

Confidence level percentile

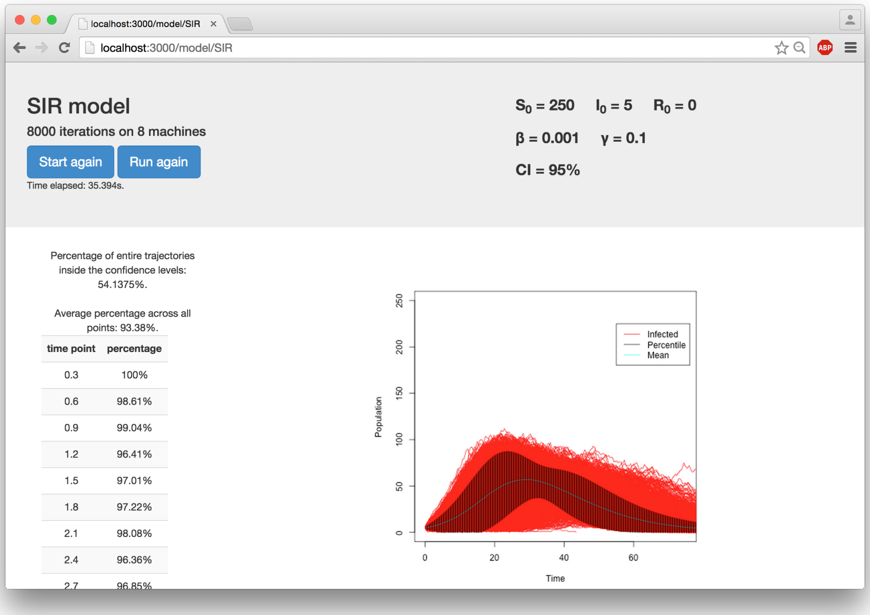
○ 85% ○ 90% ● 95%

Figure 4.3: Snapshot of parallel simulator request form

Currently, there are 3 implemented models: SIR, SIRS, SEIR. Once the desired form-tab is chosen, the user is prompted with a form personalised to the type of model chosen. For example, in the SIRS model we have an additional parameter δ, representing the rate of the recovered population that loses immunity and will join the susceptible pool. Also, for the SEIR model, we have an additional compartment of exposed individuals for which the user has to give an initial population count. In addition, the user has the following options: the time step used in the stochastic simulation, the number of machines used for simulation and the possibility of getting only the analytical calculations for the model.

Once they submit the form, the main server processes the request and splits the work accordingly, as we described above. Once the computation is finished along with the post-processing of data the user is prompted with the results of the simulation, see Figure 4.4.

This includes a plot of the simulated data, plots that illustrate the difference between the analytical and simulated mean, variance and skewness



SIR model

8000 iterations on 8 machines

Time elapsed: 35.394s.

$S_0 = 250$ $I_0 = 5$ $R_0 = 0$

$\beta = 0.001$ $\gamma = 0.1$

CI = 95%

Percentage of entire trajectories inside the confidence levels: 54.1375%.

Average percentage across all points: 93.38%.

time point	percentage
0.3	100%
0.6	98.61%
0.9	99.04%
1.2	96.41%
1.5	97.01%
1.8	97.22%
2.1	98.08%
2.4	96.36%
2.7	96.85%

Figure 4.4: Snapshot of parallel simulator response

along with statistics performed at each time point t.

From here, the user has the possibility of re-running the simulation with the exact same parameters or return to the initial form and change the parameters and/or the model. Also, we considered that the results from the simulations might be useful in a downloadable format so we made the feature available to the user.

~

Chapter 5

Evaluation

In this section we will evaluate the accuracy of the analytical moment approximations against synthetic data and the efficiency of parallelising the simulation web tool.

5.1 Moment approximation

We are going to look at the accuracy of our analytical calculations compared to the simulations for our three measures: mean, variance and skewness.

We analyse the data sets by performing unpaired t-test for each measure. The Null Hypothesis H_0 states that the mean of the two data sets are the same. We are setting the confidence level CL at 95%. Hence, for a p-value ≥ 0.05 we will accept the null hypothesis and reject it otherwise. Furthermore, we are looking at the average difference between the two data sets over the course of the epidemic.

5.1.1 Mean

We perform an unpaired t-test for our two data sets, the analytical mean and the simulated mean, at each time point t for an SIR model with parameters $S_0 = 250$, $I_0 = 5$, $R_0 = 0$, $\beta = 0.001$ and $\gamma = 0.1$ at a timestep of 0.3 and 1 for 10,000 simulations.

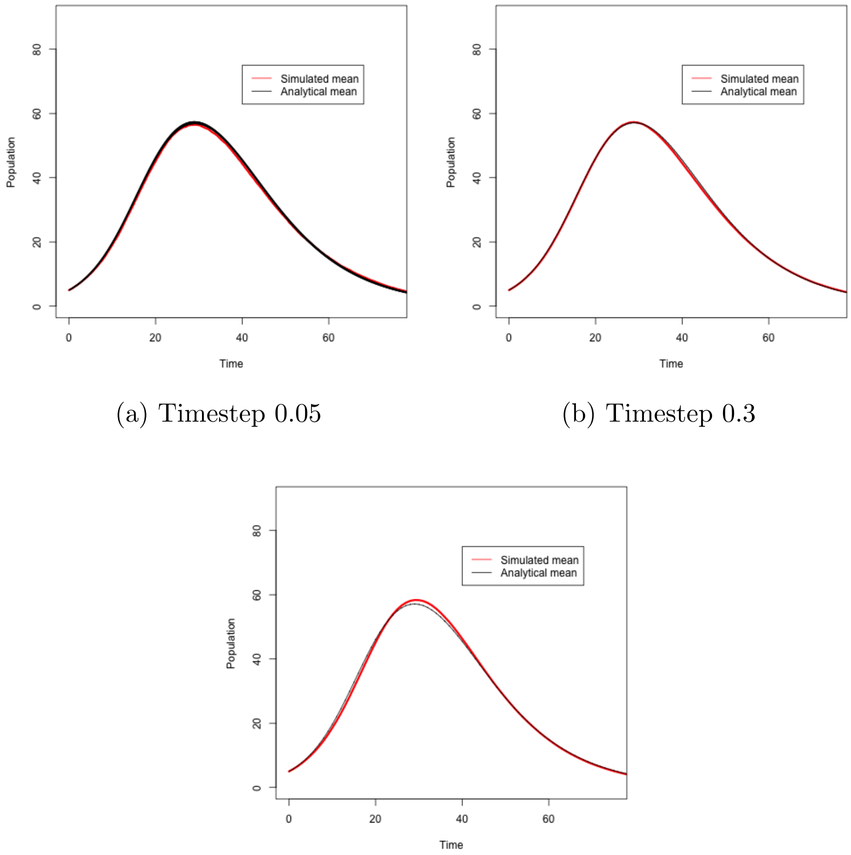
Results for a timestep of 0.3 days The average difference is -0.255. The result to the t-test is shown below:

```
        Welch Two Sample t-test

data:  mean and meanLevels
t = -0.1621, df = 537.988, p-value = 0.8713
alternative hypothesis: true difference in means
    is not equal to 0
95 percent confidence interval:
 -3.347185  2.836787
sample estimates:
mean of x mean of y
 27.40813  27.66333
```

Results for a timestep of 1 days The average difference is 0.001. The result to the t-test is shown below:

```
        Welch Two Sample t-test

data:  mean and meanLevels
t = 4e-04, df = 159.914, p-value = 0.9997
alternative hypothesis: true difference in means
     is not equal to 0
95 percent confidence interval:
 -5.777219  5.779517
sample estimates:
mean of x mean of y
 27.67039  27.66924
```

As the p-value of both unpaired t-tests is ≥ 0.05 we accept the null hypothesis, that is the means of the two data sets are the same. So, we can conclude that both methods provide the same analytical results. The interpretation of the differences observed (if any) is that they are purely due to random errors. Figure 5.1 shows the distribution of the two data sets - the analytical mean and simulated mean - for the SIR model with the parameters mentioned above, which agrees with the result of the t-test and the average difference. We note that the statistics presented above will differ slightly for each run due to the stochastic nature of our simulations.



(a) Timestep 0.05

(b) Timestep 0.3

(c) Timestep 1

Figure 5.1: Comparison between simulated mean and calculated mean at different time steps for an SIR model with parameters $S_0 = 250$, $I_0 = 5$, $R_0 = 0$, $\beta = 0.001$ and $\gamma = 0.1$

5.1.2 Variance

Similarly to Section 5.1.1, we perform an unpaired t-test for our two data sets, the analytical variance and the simulated variance, at each time point t for an SIR model with parameters $S_0 = 250$, $I_0 = 5$, $R_0 = 0$, $\beta = 0.001$ and $\gamma = 0.1$ at a timestep of 0.3 and 1 for 10,000 simulations.

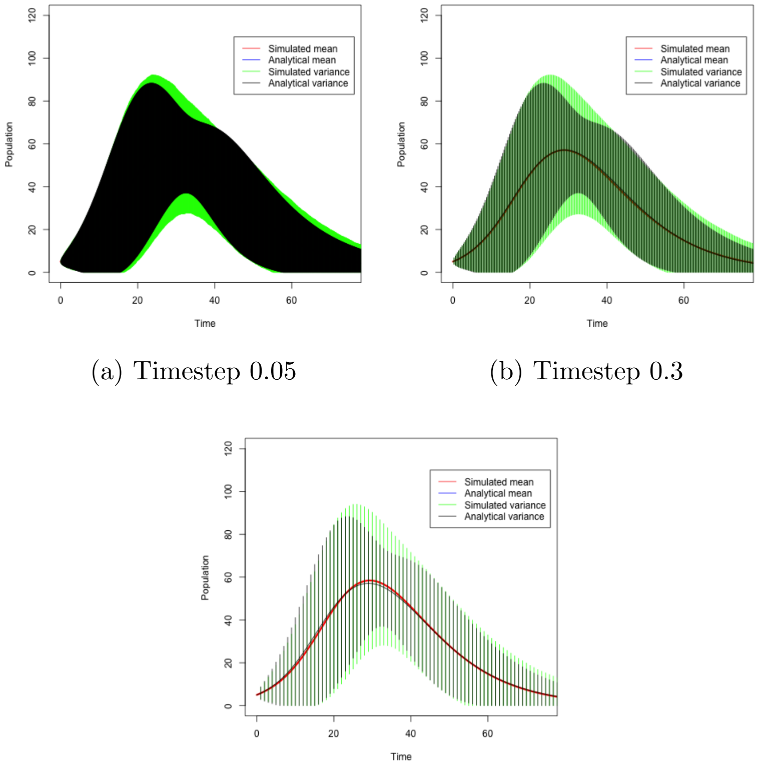
Results for a timestep of 0.3 days The average difference is 21.306. The result to the t-test is shown below:

```
        Welch Two Sample t-test

data:  varSim and varianceAnalytic
t = 2.3317, df = 532.99, p-value = 0.02009
alternative hypothesis: true difference in means
     is not equal to 0
95 percent confidence interval:
  3.356018 39.256549
sample estimates:
mean of x mean of y
 138.0449  116.7386
```

Results for a timestep of 1 days The average difference is 26.716. The result to the t-test is shown below:

```
        Welch Two Sample t-test

data:  varSim and varianceAnalytic
t = 1.5684, df = 157.519, p-value = 0.1188
alternative hypothesis: true difference in means
    is not equal to 0
95 percent confidence interval:
 -6.928338 60.361153
sample estimates:
mean of x mean of y
 143.4183  116.7019
```

The t-tests performed for the two different time steps give conflicting results. At a time step of 1 we should accept H_0, while at a time step of 0.3 we should reject H_0. Calculations are more precise with the decrease of the time step, hence we are inclined to believe the output of the t-test performed for time step 0.3. This, together with the differences observed in Figure 5.2 suggest that we should reject the Null Hypothesis. Therefore, we conclude that the analytical calculation of variance does not agree with the simulated variance. This discrepancy is introduced most likely by the approximation performed in the moment closure.



(a) Timestep 0.05

(b) Timestep 0.3

(c) Timestep 1

Figure 5.2: Comparison between simulated variance and calculated variance at different time steps for an SIR model with parameters $S_0 = 250$, $I_0 = 5$, $R_0 = 0$, $\beta = 0.001$ and $\gamma = 0.1$. The segments represent 2 standard deviations from the mean.

5.1.3 Skewness

Results for a timestep of 0.3 days The average difference is -0.019. The result to the t-test is shown below:

```
        Welch Two Sample t-test

data:  skewness and skewLevels
t = -0.327, df = 437.301, p-value = 0.7438
alternative hypothesis: true difference in means
    is not equal to 0
95 percent confidence interval:
 -0.13532919  0.09672104
sample estimates:
mean of x mean of y
0.6004772 0.6197813
```

Results for a timestep of 1 days The average difference is -0.095. The result to the t-test is shown below:

```
        Welch Two Sample t-test

data:  skewness and skewLevels
t = -0.8944, df = 134.364, p-value = 0.3727
alternative hypothesis: true difference in means
     is not equal to 0
95 percent confidence interval:
 -0.3068925  0.1157557
sample estimates:
mean of x mean of y
0.5143321 0.6099006
```

Here, we again accept the Null Hypothesis that the simulated and analytically calculated skewness are correlated. However, these results are not as strong as the ones obtained for the mean in Section 5.1.1. Again, we believe that the approximations made within moment closure are the cause for a less precise fit of the analytical skewness onto the simulated skewness.

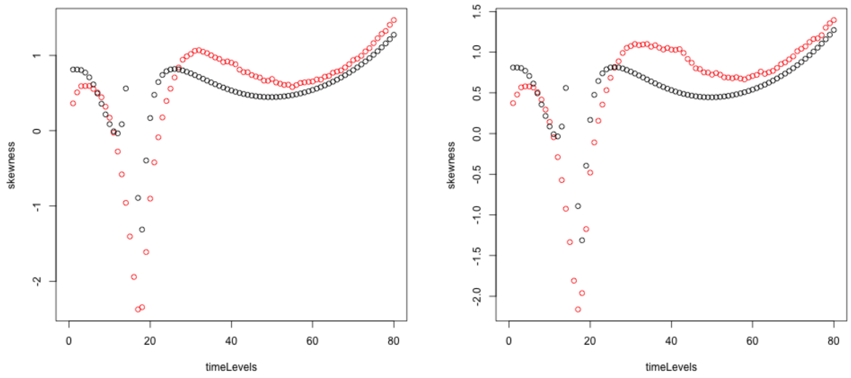


Figure 5.3: Comparison between simulated skewness and calculated skewness at time steps 1 for an SIR model with parameters $S_0 = 495$, $I_0 = 5$, $R_0 = 0$, $\beta = 0.001$ and $\gamma = 0.1$ at 10,000 runs

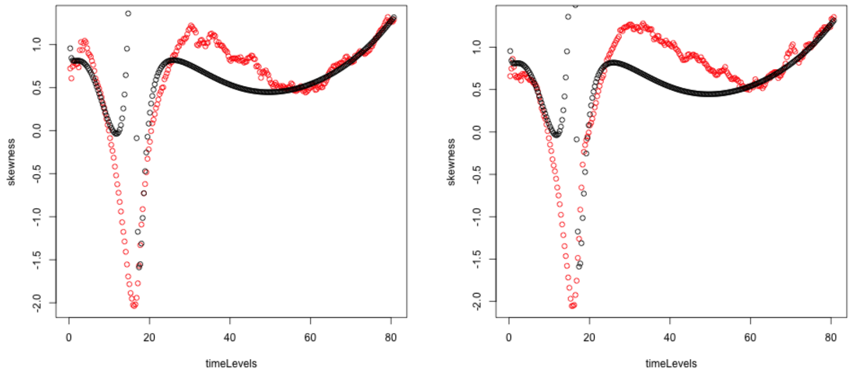


Figure 5.4: Comparison between simulated skewness and calculated skewness at time steps 0.3 for an SIR model with parameters $S_0 = 495$, $I_0 = 5$, $R_0 = 0$, $\beta = 0.001$ and $\gamma = 0.1$ at 10,000 runs

5.2 Performance

The Parallel Simulator was designed to be a time-efficient tool that can perform a large number of stochastic simulations. Below we present data regarding simulation time for different numbers of runs, different numbers of worker machines in the cloud and different simulation time steps. In addition, we present the time improvement achieved using R's parallelising capabilities.

Simulation time for different runs of the SSA algorithm In Table 5.1 we can see that the simulation time increases linearly with the number of simulations. The data was obtained using a single worker at 1,000, 10,000 and 50,000 simulation runs at a timestep of 1. From this we can approximate the time needed for n thousand simulations as n times the simulation time for 1,000 simulations.

No. of simulations	Time(s)		
	timestep = 0.05	timestep = 0.3	timestep = 1
1,000	23.9	4.12	1.49
10,000	250.2s	40.75	16.1
50,000	1205.34	210.5	85.32
n x 1,000	approx. n x 24	approx. n x 4.2	approx. n x 1.5

Table 5.1: Simulation time for an SIR model running on 8 worker machines

Simulation time for distinct sizes of the worker cloud of machines For our comparison between moments of infected counts calculated analytically and from simulations we require a large number of simulation runs to be performed. Figure 5.5 shows the time improvement achieved using different numbers of worker machines across distinct simulation time steps for 10,000 runs of stochastic simulations of an SIR model. We can also observe that the simulation time is inverse proportional with the time step. This is expected as a decrease in timestep means a proportional increase in calculations for each simulation.

R parallel package time improvement The servers used to host the Parallel Simulator are dual core machines. Hence, using R parallel package for the more intensive operations resulted, on average, in a 50% time reduction. However, we observed that the overhead of setting up the clusters is bigger than the time speed up for less intensive operations.

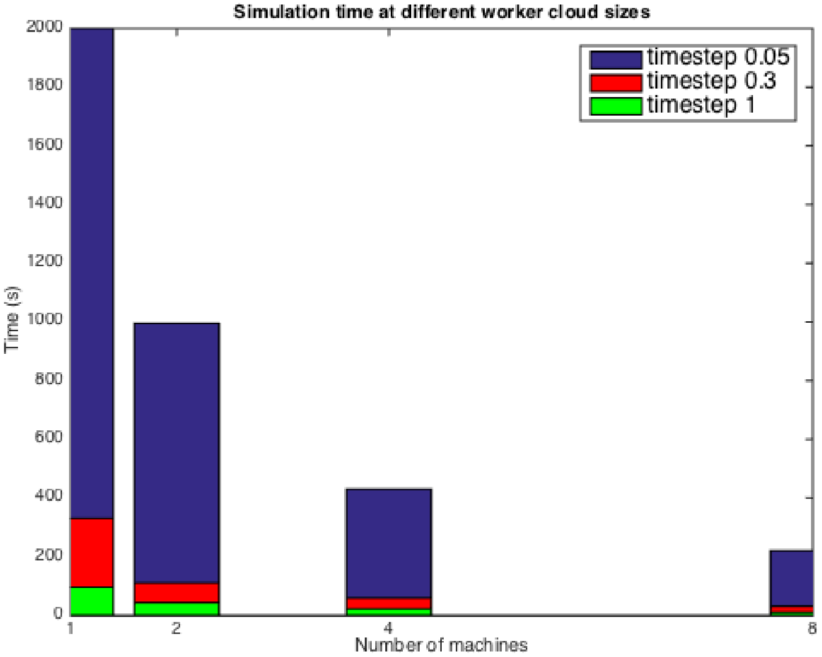


Figure 5.5: Simualtion time for different timesteps on the cloud for 10,000 simulations of the SSA algorithm

~

Chapter 6

Conclusion

For the most part, we consider this project a success. We detailed the process of analytically deriving the mean, variance and skewness of the infected counts in compartmental models and compared it against synthetic data. However, our investigations showed that the approximations made were, at times, to coarse for the aforementioned measures to correctly describe the epidemiological processes.

In addition, we successfully implemented a parallelised visualisation tool that allows for a speed-up of up to 16 times when using a cluster of 8 worker servers. The tool achieved its goal of providing users with quick realisations of stochastic simulations along with analysis and statistics of the particularities of the epidemiological model.

6.1 Future Work

In this section we discuss improvements that could be made to the work presented in this project:

- Further investigation into the errors generated by the higher-moments approximation (or otherwise) to rigorously justify the differences between analytical and simulated calculations for variance and skewness.

- Perform a comparison between different types of moment closures (normal, log-normal, min-normal etc.) to establish if they could yield better analytical moment approximations.

- Make the Parallel Simulator support user defined compartmental models. We think that the tool will become more useful if the user could define its own model through simply inputting the ODEs that define it. This would allow for additional testing of the ideas discussed in this project.

- Another extension to the simulation tool could be the ability of performing simulations in real time, allowing the user to see the output changing instantly as they modify the input parameters.

Bibliography

[1] http://www.smartglobalhealth.org/issues/entry/infectious-diseases. [Online; accessed 17-June-2015].

[2] http://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/. [Online; accessed 17-June-2015].

[3] http://www.ams.org/bookstore/pspdf/surv-176-prev.pdf. [Online; accessed 30-January-2015].

[4] http://en.wikipedia.org/wiki/Dynkin's_formula. [Online; accessed 17-June-2015].

[5] http://www.bortolussi.dmg.units.it/sites/bortolussi.dmg.units.it/files/courses/Lucca/IMT-mean-field-1.pdf. [Online; accessed 17-June-2015].

[6] http://www.bortolussi.dmg.units.it/sites/bortolussi.dmg.units.it/files/courses/Lucca/IMT-mean-field-3.pdf. [Online; accessed 17-June-2015].

[7] http://www.revolutionanalytics.com/r-community. [Online; accessed 17-June-2015].

[8] Using probabilistic models to infer infection rates in viral outbreaks. http://www.stats.ox.ac.uk/__data/assets/file/0013/3352/infection_rates.pdf, 2007. [Online; accessed 17-June-2015].

[9] http://docs.caba.org/documents/IS/IS-2014-49.pdf, 10 2013. [Online; accessed 17-June-2015].

[10] Linda J.S. Allen. Mathematical epidemiology vol. i, 2008.

[11] Nakul Chitnis. Introduction to mathematical epidemiology: Deterministic compartmental models. http://www.luchsinger-mathematics.ch/ME-DeterministicCompartmentalModels.pdf, 10 2011. [Online; accessed 17-June-2015].

[12] Klaus Dietz and J.A.P. Heesterbeek. Daniel bernoulli's epidemiological model revisited. *Mathematical Biosciences*, (180), 6 2007. [Online; accessed 17-June-2015].

[13] Murali Haran. An introduction to models for disease dynamics. 12 2009. [Online; accessed 17-June-2015].

[14] Clint Leach. Introduction to parallel computing in r. `http://michaeljkoontz.weebly.com/uploads/1/9/9/4/19940979/parallel.pdf`, 4 2014. [Online; accessed 17-June-2015].

[15] Zhien Ma. *Dynamical Modeling and Analysis of Epidemics.* World Scientific Publishing Co. Pte. Ltd., 2009.

[16] Jeremy T. Bradley Marcel C. Guenther, Anton Stefanek. Moment closures for performance models with highly non-linear rates. In *The title of the book*, volume 7587, pages 32–47. EPEW 2012, 9th European Performance Engineering Workshop, Springer, 7 2012.

[17] Ray Merril. *An Introduction to Epidemiology, Fifth Edition.* Jones and Bartlett Publishing, 2010.

[18] Marily Nika. *Synthedemic Modelling and Prediction of Internet-based Phenomena.* PhD thesis, Imperial College London, 2014.

[19] Mario Pineda-Krch. Gillespiessa: Implementing the stochastic simulation algorithm in r. *Journal of Statistical Software*, 25(12), 4 2008. `http://www.jstatsoft.org/`.

[20] Mario Pineda-Krch. Gillespie's stochastic simulation algorithm (ssa). `http://cran.r-project.org/web/packages/GillespieSSA/GillespieSSA.pdf`, 2 2015. [Online; accessed 30-January-2015].

[21] Mark Pollicott, Hao Wang, and Howie Weiss. Extracting the time-dependent transmission rate from infection data via solution of an inverse ode problem. `http://people.math.gatech.edu/~weiss/pub/PWWrevisionJBD.pdf`. [Online; accessed 17-June-2015].

[22] Anton Stefanek. *A high-level framework for efficient computation of performance-energy trade-offs in Markov population models.* PhD thesis, Imperial College London, 2014.