

Stochastic Modelling of Malaria

Ben Lowery

Supervisor: Dr. Jie Yang

Abstract

By considering a compartmental model for the spread of malaria, this thesis aims to evaluate the effectiveness of stochastic models in the spread of infectious diseases. This is achieved through introducing different variations and avenues which can be taken with stochastic modelling of the spread of disease by delving into the evolution of discrete and continuous times models that employ randomness to account for the uncertainty that is prevalent in real world cases. Models range from simple discrete-time Markovian processes, to implementing theory from stochastic calculus to create stochastic differential equations from underlying ordinary differential equations. A sensitivity analysis on the various models is provided. Key characteristics for disease models such as the basic reproduction number and the disease free equilibrium are explored. A model validation is included using a dataset concerning cases in an endemic region of Malaysia. Finally, the thesis culminates in a discussion and conclusion on the practicalities, utility and viability of stochastic compartmental disease modelling in comparison to other techniques employed in both historic and contemporary times.

Contents

1	Introduction	5
1.1	Historical overview of infectious disease modelling	5
1.2	How diseases are modelled	6
1.2.1	Stochastic models	8
1.3	Mitigating the spread of diseases	9
1.4	Malaria modelling	10
2	Preliminaries	12
2.1	Epidemiology	12
2.1.1	Deterministic SIR model	12
2.1.2	Next-generation matrix	13
2.2	Probability and Statistics	14
2.3	Itô Calculus	16
3	Methodologies	18
3.1	Chain Binomial model (Reed-Frost)	18
3.2	Branching Processes	20
3.3	Stochastic Differential Equations	23
3.3.1	Derivation	23
3.4	Simulation of Epidemic Models	24
3.4.1	Gillespie's Algorithm	25
3.4.2	Numerical methods for SDEs	26
3.4.3	Convergence approximations of numerical SDE methods	28
4	A stochastic model for the spread of malaria	29
4.1	Deterministic model and properties	31
4.1.1	Disease-free equilibrium	31
4.1.2	Basic reproduction number	32
4.2	Stochastic vector-borne model	33
4.3	Parameter choices	35
4.3.1	Sensitivity Analysis: Overview	36
4.4	Model adaptations	38
4.4.1	Variable mosquito population size	38
4.4.2	Introduction of Wolbachia to the population	40
5	Results	43
5.1	Baseline SIR models	43
5.1.1	Chain Binomial Model	43
5.1.2	Stochastic simulation with Gillespie's Algorithm	44
5.1.3	SDE numerical simulation	45
5.2	Malaria Models	46
5.2.1	Main malaria model	46
5.2.2	Malaria models with variable mosquito population	49

5.2.3	Malaria model with introduction of Wolbachia	52
6	Sensitivity Analysis: Results	53
6.1	Sensitivity of \mathcal{R}_0	53
6.1.1	Main malaria model	54
6.1.2	Adapted models (doubling population)	55
6.2	Sensitivity of other properties	57
6.2.1	Loss of immunity	57
6.2.2	Wolbachia rate	58
7	Model Validation	59
8	Discussion	63
9	Conclusion	66
10	Future directions	67
	References	69
	Appendices	75
	A Gillespie Algorithm Variables	75
	B Code listings	76

Nomenclature

Population Dynamics

H Total human population

K Carrying capacity

M Total mosquito population

N Total population

Model Parameters

α_1 Rate of infection from human to mosquito

α_2 Rate of infection from mosquito to human

β Infection rate

γ Recovery rate

μ Natural death rate of mosquitoes

ω Depletion rate from Wolbachia

θ Per capita rate for loss of immunity

k Bite rate of mosquito

p Probability of infection from mosquito to human from bite

q Probability of infection from human to mosquito from bite

1 Introduction

1.1 Historical overview of infectious disease modelling

“I simply wish that, in a matter which so closely concerns the well-being of the human race, no decision shall be made without all the knowledge which a little analysis and calculation can provide.”

- Daniel Bernoulli, 1760

In 1766, Swiss Mathematician Daniel Bernoulli published an article in the French literary magazine *Mercure de France* concerning the effect smallpox had on life expectancy and the improvements which could be made with the introduction of inoculation. Bernoulli’s concluding argument, from which the aforementioned quote is derived, led to the creation of some of the first epidemiological models. Contemporary literature by [Dietz and Heesterbeek \(2002\)](#) and [Bacaër \(2011a\)](#) provide insight into the motivation and subsequent impact the article had on modelling infectious diseases alongside methods to mitigate the impact of a disease such as smallpox. This does not mean that the paper by Bernoulli was immediately lauded by his peers. Jean le Rond d’Alembert, a prominent mathematician and intellectual at the time (albeit slightly behind the curve in probability theory¹) clashed with Bernoulli on the issue, authoring a rebuttal in a 1760 paper *On the application of probability theory to the inoculation of smallpox*. Bernoulli had initially communicated his work in a 1760 presentation in Paris (see [Bacaër, 2011a](#)). However, issues caused the paper to not be released until 1766, giving d’Alembert a head start in his critique. He argued against some of the assumptions Bernoulli had made with respect to the probability of infections and the independence between age and dying of smallpox. D’Alembert’s alternative formulation also resembles modern modelling and, despite the differing opinions, both agreed that inoculation of the population was the way forward.

Bernoulli’s submission is often regarded as the foundation to what would eventually become mathematical epidemiology; although the field as it is known today did not further develop until the early 20th century. This arrived in the form of work by polymath Sir Ronald Ross, who wrote on malaria prevention by crafting a model of two differential equations ([Ross, 1910](#)). More generalised models were also produced by [Kermack et al. \(1927, 1932, 1933\)](#), which provided early forms of compartmental modelling; a class of models that form the cornerstone of much subsequent research. These models took a deterministic form where no randomness is involved in the system and the output will always be replicated if given the same set of initial parameters. These were defined as a system of ordinary differential equations (ODEs).

Another class of compartmental models that arrived not too long after deterministic models were stochastic models. These types of models included the effects of randomness commonly found in real-life scenarios. One of the earliest stochas-

¹Famously in his paper “Croix ou Pile”, d’Alembert (incorrectly) reasoned that the odds of attaining a head in a two toss game was $\frac{2}{3}$ ([Daston, 1979](#)).

tic models to appear was the Reed-Frost model. Introduced in a 1928 lecture by [Frost \(1928\)](#)- but not formalised until the 1950s (see [De Oliveira Costa Maia, 1952](#)) - alongside a deterministic version, the stochastic model employed discrete time dynamics and provided a simple formulation of disease spread. Other, more complex, stochastic model classes include using continuous time dynamics and stochastic differential equations (SDEs) ([Allen, 2008](#)).

Another substantial movement in the field came with the landmark paper by [Anderson and May \(1981\)](#), followed by the book *Infectious Diseases of Humans: Dynamics and Control* ([Anderson and May, 1992](#)). In these seminal works, the authors created models based on physical observations and data, with a strong emphasis on the biological assumptions made. They also paved the way for collaboration that combined different disciplines, all sharing an interest in modelling diseases. The two are, rather emphatically in a review article by [Heesterbeek and Roberts \(2015\)](#), credited to providing a pivotal shift in the field, invoking a more integrated relationship between Applied Mathematics and Biology.

More recently, agent-based (or individual) models, took a step away from compartmentalising diseases in populations, providing an alternative focus on the creation of more complex systems to accurately reflect real world situations. Compartmental models are often simplified and restricted by the mathematics used to construct them ([Railsback and Grimm, 2011](#)), thus agent-based models, henceforth denoted ABMs, are based around using computational simulations to replicate interactions between populations or ‘agents’. Due to these complexities and the computational capacity needed to run ABMs, progress in the field was slow until the 1990s. During the COVID-19 pandemic, they have heavily influenced policy decisions, with Neil Ferguson’s COVIDSim model ([Ferguson, 2020](#)) being used to aid the UK government in its response throughout the pandemic ([Adam, 2020](#)). Despite the more complex systems ABMs can create, criticisms have been made on their over reliance, and simplification, of real world systems ([Sridhar and Majumder, 2020](#)).

1.2 How diseases are modelled

Infectious diseases can be classified into different sub-types. Focusing on diseases affecting a certain region or population group, the terms epidemic and endemic can be used in certain scenarios. If the disease is spreading rapidly through a population over a short period of time, it is then classified as an epidemic. However, if the disease is always persistent in a population, as opposed to complete extinction, the disease is then in a steady state and endemic.

When modelling infectious diseases, researchers, contemporary and historic, can use a number of methods to simulate spread through a population. As recently touched upon, these can take the form of compartmental models (deterministic and stochastic) or Agent-Based models; the former of which will be the area of focus.

Trivially put, compartmental models are made of different ‘compartments’ where a population is distributed to. A commonly used model throughout literature splits a population into three stages: Susceptible, Infected, and Removed

(this can also be denoted as recovered individuals where the terms are interchangeable and usually dependent on whether the disease has fatal or non-fatal repercussions), overall referenced as an **SIR** model. Other compartments that can expand the model, include the introduction of **Deceased**, **Maternally-derived immunity** or **Exposed** groups; however these are greatly dependent on the disease and consideration should be given regarding whether these population groups are negligible or not.

The simplest way to introduce the SIR model would be through its deterministic formulation with a replication of the Kermack-McKendrick model ([Kermack et al., 1927](#)). To begin, the following basic assumptions about the SIR model can be made: (i) The population is fixed; (ii) The disease can only be spread between members within the population; (iii) No spatial or demographic considerations are accounted for; (iv) the length of the infectious period is the same as the length of the disease.

Functions denoting the number of people susceptible, infected or recovered at time t are given by $S(t)$, $I(t)$, and $R(t)$, respectively. A system of coupled nonlinear ODEs can be expressed by Equations (1) to (3):

$$\frac{dS}{dt} = -\beta SI, \quad (1)$$

$$\frac{dI}{dt} = \beta SI - \gamma I, \quad (2)$$

$$\frac{dR}{dt} = \gamma I. \quad (3)$$

Here, β is the infection rate and γ the recovery rate. A key value to come from this model is the basic reproduction number, \mathcal{R}_0 . Defining the number of secondary infections that come as a result of an infectious individual, succinctly, this value is expressed as,

$$\mathcal{R}_0 = \frac{\beta}{\gamma} \quad (4)$$

for the Kermack-McKendrick model. If \mathcal{R}_0 is below 1, then the disease will not be able to spread in the population. A value of \mathcal{R}_0 greater than 1 signifies a rapid spread of the disease and an epidemic arising. Finally, a \mathcal{R}_0 equal to 1 yields an endemic state and a persistent disease. This model is restrictive and simplistic in its assumptions and formulation. Nevertheless, it provides a useful introduction and foundation to compartmental modelling. A formal definition of a SIR model, analogous to the one just stated, can be found in Section [2.1.1](#).

The focus so far has been on the spread of a contagious disease solely through the members of a single population group. However, with certain diseases, the contagious aspect might not be present. Instead, the disease is spread from separate organisms to a susceptible population. Vectors can be defined as organisms that carry an infectious pathogen which infect a population and give rise to “vector-borne diseases”. The contagious aspect may not be present and the only way of infection would then be from the vector to the susceptible individual. Modelling such scenarios requires an adaption of the compartmental models, whether this be

through the introduction of more ODEs to represent the vector population or an addition of extra parameters to incorporate infected vectors. The SIR-SI model is one possible avenue using the former approach, and has been used to model dengue fever in work by [Pandey *et al.* \(2013\)](#). A schematic diagram of the models dynamics can be seen in Figure 1.

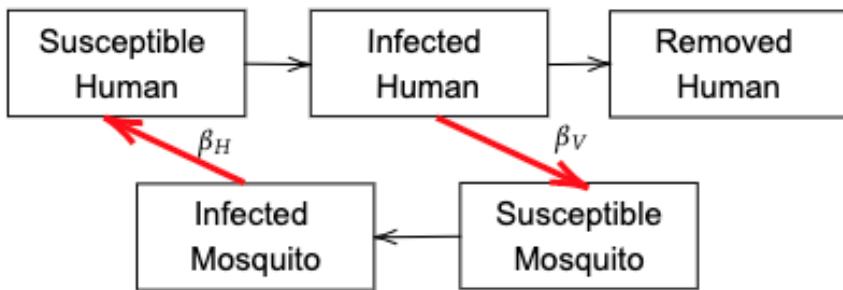


Figure 1: Schematic diagram of SIR-SI model for transmission of Dengue. The red arrows represent transmission between human and mosquito interaction, with β_V and β_H denoting the transmission probability for human-to-mosquito and mosquito-to-human respectively.

1.2.1 Stochastic models

Deterministic models are not the only tools for modelling epidemics with stochastic models providing an alternate choice. One such motivation behind the needs of such techniques is monitoring the early stages of an epidemic. Models like Kermack-McKendrick rely on the population for each compartment to be large and well mixed, so any one member will more than likely infect the same amount as another. This clearly cannot be the case at the beginning of epidemics with low numbers of infectious individuals. Therefore, stochasticity can be introduced and a distribution that can scale and reflect infections patterns at various stages. To model the beginning of epidemics in particular, a branching process can be used. More explicitly, an early stage epidemic can be suitably described by a set of vertices on a graph representing members of a population with edges connecting these. Numerous approaches can then be used to describe a transition rate to which an initial infectious individual (“patient zero”) can infect nearby vertices; see [Brauer *et al.* \(2019\)](#) for a brief overview and extra motivation behind using branching processes for early stage outbreaks.

When it comes to the main stages of an epidemic, control methods can be implemented in order to reduce the spread of a disease actively prevalent in a community. If an epidemic is prevalent in the community, a shift back to compartmental modelling is likely warranted; with these compartmental models not necessarily having to be deterministic. Three popular types of stochastic models can be considered in this case: discrete-time models, continuous-time models and stochastic differential equations (SDE). Stochastic models help accommodate for the random and imperfect nature of how diseases evolve in real world environments, capturing possible outcomes deterministic models cannot. In contrast,

qualitative results are much more fruitful in deterministic models. Stochastic models can be extremely useful in capturing the uncertainty pertaining to the spread of infectious diseases and provide useful quantitative results in predicting outcomes of epidemics. A primer on different stochastic models of varying complexity is explored in Section 3, but there are a plethora of methods that extend far beyond the scope of this thesis. These are touched upon in the concluding sections.

1.3 Mitigating the spread of diseases

One challenge to be carefully considered is the introduction of events and policies that can mitigate and impact how a disease spreads. These can be split into **Prevention** and **Control** categories. Both have numerous tactics and methods to prevent the spreading of an infectious disease, and reduce the possibility of an epidemic occurring. These policies can also be incorporated into mathematical models, with varying degrees of simplicity and complexity available to be incorporated. In deterministic compartmental models, these can be added as extra parameters to different stages and integrated to reflect a slower transition of people from one state to another.

Focusing on specific measures, prevention techniques can be employed to stop the transition between groups. For example, primary prevention techniques are targeted at susceptible groups and individuals with the aim of completely preventing the disease from spreading in the first place. Immunisation and the needle exchange program (used for the prevention of blood-borne diseases such as HIV and Hepatitis) can be effective implementations of primary prevention ([Kisling and Das, 2021](#)). If there are limited or no forms of inoculation, non-pharmaceutical interventions (NPIs) can be brought in to stem the spread. Quarantine, contact tracing, restriction of movement and introduction of personal protection equipment (PPE) for health workers are examples of such methods. Given suitable data, mathematical models can gauge the effects these measures can have; say for spatial models, the introduction of movement restrictions can be implemented with the model gaining extra parameters that can restrict movement between areas.

In practice, compliance to measures is not a guarantee. Low government trust, “lockdown fatigue” and vaccine hesitancy can negatively impact effectiveness of preventive and control measures. As such, models can struggle to reflect these intricacies, whether this is caused by incorrect assumptions by modellers or the sheer gap in complexity between real life and mathematical representation, these issues can have large consequential effects on the effectiveness of a model.

Preventative measures can also be introduced to mitigate the spread of vector-borne diseases. One example, used in particular to combat *Aedes mosquitoes* carrying the Zika virus, is the introduction of mosquitoes carrying the bacterium *Wolbachia* and release of insects with dominant lethality (RIDL). These allow for a targeted, specific species approach, alongside a low environmental pollution cost. Introduction of such measures, however, have been restricted by government funding and resource allocation ([Dickens et al., 2016](#)). Other traditional methods for vector-borne diseases include insecticides and environmental controls ([Wilson](#)

et al., 2020).

1.4 Malaria modelling

The primary motivation of this thesis is to produce and examine a model for simulating the spread of the vector-borne disease malaria. Spread by infected mosquitoes predominately carrying the parasite *P. falciparum*, the World Health Organisation estimated 13 million cases in the Eastern Mediterranean Region alone in 2014 (WHO, 2014). The disease has potentially dangerous consequences with 10% of cases caused by the falciparum parasite being classed as severe and 10% of those proceeding to be fatal; this severe proportion can reach as high as 50% depending on case definition (Pasvol, 2006). Malaria is also heavily distributed to poorer regions, giving rise to potential issues such as under-reporting and lack of effective resources to combat the disease (Snow *et al.*, 2005).

Sir Ronald Ross produced a much commended mathematical model of malaria. He concluded his work with an argument that a reduction of mosquitoes in the population would lead to the eradication of malaria. To reach this outcome, Ross (1910) formulated a set of two differential equations, representing the number of infected individuals in the mosquito and human population, and subsequently analysed the steady states. Focusing on the steady state in which malaria was present and endemic in the population, Ross' findings were that if the number of mosquitoes was to fall below a critical threshold, the only remaining steady state is disease free. Therefore, a reduction, not extinction, of the mosquito population was sufficient for eradication (Bacaër, 2011b).

Following on from the work of Ross, Macdonald (1956) extended the analysis, as a result spurring on the creation of the “Ross-MacDonald” model. Notable features to develop from this, in addition to the idea of a critical threshold, was the non-linear relationship when it comes to the prevalence of infection between human and mosquito populations, and the basic reproduction number. Due to the positive equilibrium for the reproduction rate above 1, the instigation of temporary measures to lower the infection rate will itself only have a temporary effect. Relaxing such measures would simply cause infections to rise back to previous values (Ruan *et al.*, 2008). An important addendum to information regarding the model is that Ross-MacDonald does not have a set defined model, rather numerous models encapsulated under the banner that all share a consensus set of assumptions. Highlighted by Smith *et al.* (2012), the theory has had a pivotal role in the research on mosquito transmission.

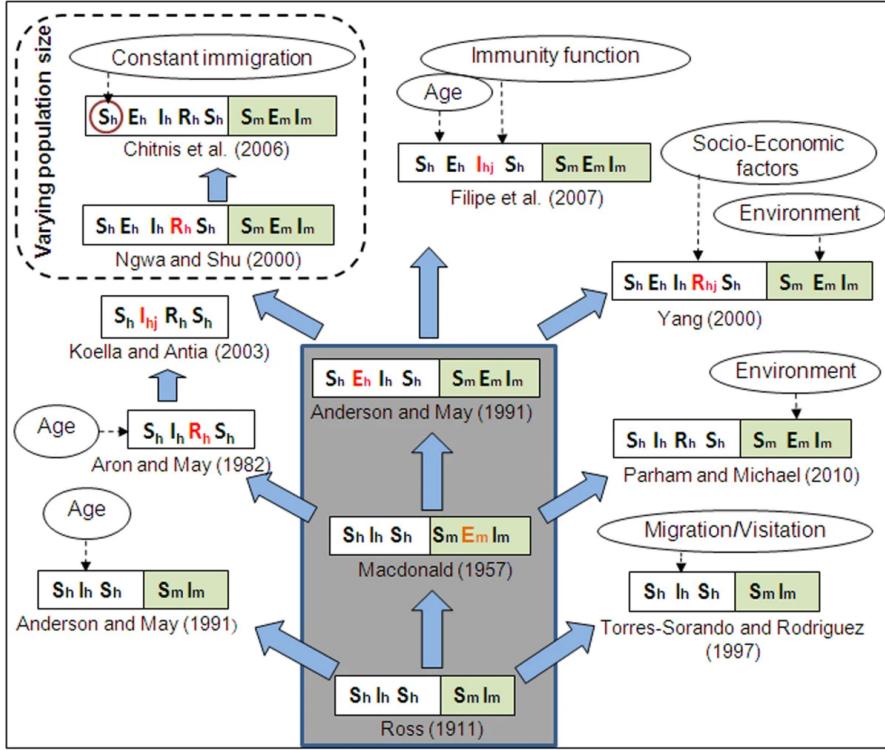


Figure 2: Evolution of different deterministic malaria models courtesy of [Mandal et al. \(2011\)](#). Highlighting the evolution of epidemic models, stemming from Ross and Macdonald, here h denotes human population and m mosquito population. It can be seen that recent models adopt variations of an SEIRS compartmental set up, and most models consider separate human and mosquito dynamics in the form of a vector-borne model.

Ross-MacDonald models are of a deterministic form and these dominate the literature due to their simplicity and the extensive information already available. However, randomness and stochastic elements have also been actively included into deterministic malaria models ([Mandal et al., 2011](#)). Some adaptations and extensions of Ross-MacDonald models can be seen in Figure 2 with factors such as demography and Socio-Economic considerations being added to the baseline model. With regards to purely stochastic models, these have been actively researched in a smaller subsection of the literature. Statistical fluctuations regarding infection behaviour has led to researches adopting a stochastic simulation approach ([Smith et al., 2008](#)). [Molineaux et al. \(2001\)](#) proposed a discrete time model fitted to 35 historical case studies. [Gatton and Cheng \(2004\)](#) carried out stochastic simulations to model acquired clinical immunity in individuals. A stochastic differential equation set up, modelling spatial spread across a lattice, was recently proposed by [Le et al. \(2018\)](#). The stochastic lattice-based integrated malaria (SLIM) model, derives itself from a deterministic formulation by [Lutambi et al. \(2013\)](#). A deterministic model can provide apt groundwork for building up stochastic variants and is a process to be followed in this thesis.

2 Preliminaries

2.1 Epidemiology

Proposition 2.1 (Epidemiological Law of Mass Action). *For a number of susceptible individuals S , and infectious individuals I , the rate at which an infection passes in a population is proportional jointly to the product of S and I .*

Definition 2.1 (Basic Reproduction Number). The number of secondary infections resulting from a single primary infection into an otherwise wholly susceptible population, denoted \mathcal{R}_0 .

2.1.1 Deterministic SIR model

Let $S(t)$, $I(t)$, $R(t)$ denote the number of susceptible, infected, and removed individuals respectively, at time t . A simple deterministic SIR model without births or deaths is given in Equation (5) as follows:

$$\begin{aligned}\frac{dS(t)}{dt} &= -\beta \frac{S(t)}{N} I(t), \\ \frac{dI(t)}{dt} &= \beta \frac{S(t)}{N} I(t) - \gamma I(t), \\ \frac{dR(t)}{dt} &= \gamma I(t).\end{aligned}\tag{5}$$

Here, $S(t), I(t), R(t) \geq 0$ and β, γ denotes the infection and recovery rate, respectively. The model also assumes a constant population size with $S + I + R = N$. The basic reproduction number of this model is expressed as,

$$\mathcal{R}_0 = \frac{\beta}{\gamma}.\tag{6}$$

When $S(0) \approx N$ and $I(0) > 0$, Equation (6) can be used to describe the dynamics of the model in the following theorem.

Theorem 2.2. *Let $S(t)$, $I(t)$, and $R(t)$ be solutions to Equation (5). Then,*

1. *If $\mathcal{R}_0 S(0)/N > 1$, then the number of infectious individuals $I(t)$ increases and begins an epidemic process.*
2. *If $\mathcal{R}_0 S(0)/N < 1$, then the number of infectious individuals $I(t)$ monotonically decreases to zero and eventually reaches a disease free equilibrium.*

Note, since $S(t) + I(t) + R(t) = N$, this implies $R(t) = N - S(t) - I(t)$ which reduces Equation (5) to a system of two equations if required.

2.1.2 Next-generation matrix

Suggested initially by Diekmann *et al.* (1990) and expanded on in the work by van den Driessche and Watmough (2002), the next generation matrix is an approach to calculate the basic reproductive number and is primarily used for more complex compartmental models. The overall approach to calculate the next-generation matrix is split into three stages:

Step 1

The population is split into n compartments, containing within it, m infection compartments ($m < n$). Denote x_i for $i = 1, 2, \dots, m$ as the number of infected individuals in an infected compartment i at time t . These compartments are now given as,

$$\frac{dx}{dt} = \mathcal{F} - \mathcal{V}. \quad (7)$$

In Equation (7), \mathcal{F} is a $m \times 1$ vector containing the rates for the introduction of new infections in the respective compartment, whilst $\mathcal{V} = \mathcal{V}^- - \mathcal{V}^+$ is the $m \times 1$ vector for the rate of transfer out of each respective compartment (\mathcal{V}^-) and the transfer in of individuals by other means (\mathcal{V}^+).

Step 2

Construct $m \times m$ matrices \mathbf{F} and \mathbf{V} of the partial derivative of each value in \mathcal{F} and \mathcal{V} respectively, with respect to the infected compartment x_i . More specifically,

$$\mathbf{F} = \frac{\partial \mathcal{F}_i}{\partial x_j}, \quad \mathbf{V} = \frac{\partial \mathcal{V}_i}{\partial x_j}. \quad (8)$$

In Equation (8), \mathcal{F}_i and \mathcal{V}_i are the corresponding i^{th} vector values to the respective matrices.

Step 3

Evaluating at the disease free equilibrium \bar{x} , the matrix $\mathbf{F}\mathbf{V}^{-1}(\bar{x})$ is known as the next-generation matrix. To recover the basic reproduction number from the next-generation matrix, the largest eigenvalue needs to be found. The function that recovers the largest eigenvalue can be formally defined as the Spectral Radius.

Definition 2.2 (Spectral Radius). Let $\lambda_1, \lambda_2, \dots, \lambda_n$ be eigenvalues of a $n \times n$ matrix \mathbf{A} . Then, the spectral radius, $\rho(\mathbf{A})$, is defined as:

$$\rho(\mathbf{A}) = \max\{|\lambda_1|, \dots, |\lambda_n|\}. \quad (9)$$

2.2 Probability and Statistics

The mathematics in this section is derived, explicitly or adapted, from the textbook *Probability and Statistics* (DeGroot and Schervish, 2012).

Definition 2.3 (Stochastic Process). A sequence of random variables, $\{X(t)\}_{t \in T}$ is called a stochastic process or random process. A stochastic process can be discrete or continuous depending on whether the set T is countable or on an interval.

Definition 2.4 (Markov Property). A discrete-time Markov Chain on a finite (or countable) set is a stochastic process $\{X_n | n = 0, 1, 2, \dots\}$ with values in the set, such that

$$\mathbb{P}[X_{n+1} \leq b | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n] = \mathbb{P}[X_{n+1} \leq b | X_n = x_n]. \quad (10)$$

This can also be referred to as the *Memoryless property*.

Definition 2.5 (Binomial Distribution). A random variable X is said to have the binomial distribution, with parameters n (the number of trials) and p (probability of success), if X has a discrete distribution with the following probability function:

$$f(x|n, p) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x}, & \text{for } x = 0, 1, \dots, n \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

A random variable with this distribution can then be denoted $X \sim Bin(n, p)$.

Definition 2.6 (Bernoulli Distribution). A random variable X that only takes the values 0 and 1, with $\mathbb{P}[X = 1] = p$, has the Bernoulli distribution with parameter p .

Definition 2.7 (Bernoulli Trial). For a sequence of random variables X_1, X_2, \dots that are i.i.d., with the Bernoulli distribution, the random variables are said to be Bernoulli trials with parameter p . With an infinite sequence of Bernoulli trials, it is said to form a Bernoulli process.

Definition 2.8 (Poisson Distribution). Let $\lambda > 0$. A random variable, X , is said to have the Poisson distribution with parameter λ (the mean), if X has the probability function:

$$f(x|\lambda) = \begin{cases} \frac{e^{-\lambda}\lambda^x}{x!} & \text{for } x = 0, 1, 2, \dots, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

A random variable with this distribution can then be denoted $X \sim Pois(\lambda)$.

Definition 2.9 (Normal Distribution). A random variable X is said to have a normal distribution, with mean μ and variance σ^2 , if X has a continuous distributions with the following probability density function:

$$f(x|\mu, \sigma^2) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \text{ for } -\infty < x < \infty. \quad (13)$$

A random variable with this distribution can be denoted $X \sim Normal(\mu, \sigma^2)$.

Theorem 2.3 (Central Limit Theorem). *Let X_1, X_2, \dots, X_n be independent and identically distributed (i.i.d) random variables of size n from a given distribution with mean μ and variance σ^2 . Then, for each fixed number x :*

$$\lim_{n \rightarrow \infty} \mathbb{P} \left[\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \leq x \right] = \Phi(x), \quad (14)$$

where Φ denotes the cumulative distribution function (cdf) of the standard normal distribution.

Theorem 2.4 (Multivariate Central Limit Theorem). *Let X_1, X_2, \dots, X_n be i.i.d random variables with mean μ and covariance matrix Σ , then, $(\bar{X}_n - \mu)/\sqrt{n}$ converges to a Normal distribution, $\text{Normal}(0, \Sigma)$.*

Definition 2.10 (Probability Generating Function). Let X be a random variable with non-negative integer values. Then, the probability generating function (pgf) of X is given as,

$$G_X(t) = \mathbb{E}[t^X] = \sum_{j=0}^{\infty} \mathbb{P}[X = j] t^j = \sum_{j=0}^{\infty} p_j t^j. \quad (15)$$

Probability generating functions also satisfy the following properties:

- (i) $G_X(0) = \mathbb{P}[X = 0]$,
- (ii) $G_X(1) = 1$,
- (iii) The probability mass function of X is attained through the derivatives of the pgf, specifically,

$$p_n = \mathbb{P}[X = n] = \frac{G^n(0)}{n!}.$$

2.3 Itô Calculus

This section references the lecture notes on stochastic finance by [Privault \(2015\)](#).

Definition 2.11 (Wiener Process). A one-dimensional standard Wiener process (also known as Brownian Process) is a stochastic process $\{W(t)\}_{t \in \mathbb{R}^+}$ such that:

- (i) $W(0) = 0$.
- (ii) The function $t \mapsto W(t)$ is continuous on t , with probability 1.
- (iii) For any finite sequence of time, $t_0 < t_1 < \dots < t_n$, the increments $W_i(t) - W_{i-1}(t)$ are mutually independent random variables.
- (iv) For any given time $0 \leq s < t$, $W(t) - W(s) \sim \text{Normal}(0, t - s)$.

Note that this can be extended to a n-dimensional Wiener process with the vector:

$$\mathbf{W}(t) = [W_1(t) \ W_2(t) \ \dots \ W_n(t)]^T,$$

where $W_i(t)$ are independent one-dimensional Wiener processes.

Theorem 2.5 (Itô's Uniqueness Theorem). *Define functions,*

$$f : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$$

and

$$\sigma : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}^d \otimes \mathbb{R}^n,$$

where $\mathbb{R}^d \otimes \mathbb{R}^n$ denotes the space of $d \times n$ matrices. Let both functions satisfy the global Lipschitz condition and assume there exists a $K > 0$ such that,

$$\|\sigma(t, x) - \sigma(t, y)\|^2 + \|f(t, x) - f(t, y)\|^2 \leq K^2 \|x - y\|^2, \quad (16)$$

with $t \in \mathbb{R}_+$, $x, y \in \mathbb{R}^n$. Then, there exists a unique strong solution to the stochastic differential equation:

$$X(t) = X(0) + \int_0^t f(X(s))ds + \int_0^t \sigma(X(s))dW(s), \quad t \geq 0. \quad (17)$$

The proof of Itô's uniqueness theorem is an application of a fixed point theorem in what is known as a Banach Space (a complete normed vector space). An outline of the proof can be briefly split into three parts: first is to show the stochastic process $\{X\}_{t \in T}$ is a member of a predefined Banach space, then show existence of solutions on the time horizon T , and finally apply this to future time intervals $[T_n, T_{n+1}]$. A complete proof is provided by [Baudoin \(2019\)](#).

Definition 2.12 (Itô SDE). Writing (17) in differential notation gives the general form of a Stochastic Differential Equation (SDE):

$$dX(t) = f(X(t))dt + \sigma(X(t))dW(t). \quad (18)$$

With stochastic processes $\{W(t)\}_{t \in \mathbb{R}_+}$ and $\{X(t)\}_{t \in \mathbb{R}_+}$.

Definition 2.13 (Itô Taylor Expansion). For sufficiently smooth functions f and σ , the stochastic Taylor expansion is given as,

$$\begin{aligned} X(t) &= X(t_0) + f(X(t_0)) \int_{t_0}^t d\tau_1 + \sigma(X(t_0)) \int_{t_0}^t dW(\tau_0) \\ &\quad + \sigma(X(t_0))\sigma_x(X(t_0)) \left[\frac{1}{2}[W(t) - W(t_0)]^2 - \frac{1}{2}(t - t_0) \right] + \tilde{\mathcal{R}}, \end{aligned} \quad (19)$$

where $\sigma_x(X(t))$ is the partial derivative of σ with respect to $X(t)$. In addition, $\tilde{\mathcal{R}}$ is the remainder term expressed as,

$$\begin{aligned} \tilde{\mathcal{R}} &\equiv \int_{t_0}^t \int_{t_0}^{\tau_1} L^0 f(X(\tau_2)) d\tau_2 d\tau_1 + \int_{t_0}^t \int_{t_0}^{\tau_1} L^1 f(X(\tau_2)) dW(\tau_2) d\tau_1 \\ &\quad + \int_{t_0}^t \int_{t_0}^{\tau_1} L^0 \sigma(X(\tau_2)) d\tau_2 dW(\tau_1) \\ &\quad + \int_{t_0}^t \int_{t_0}^{\tau_1} \int_{t_0}^{\tau_2} L^0 L^\sigma(X(\tau_3)) d(\tau_3) dW(\tau_2) dW(\tau_1) \\ &\quad + \int_{t_0}^t \int_{t_0}^{\tau_1} \int_{t_0}^{\tau_2} L^1 L^1 \sigma(X(\tau_3)) dW(\tau_3) dW(\tau_2) dW(\tau_1) \\ &\quad + \text{Higher Order Terms.} \end{aligned} \quad (20)$$

In $\tilde{\mathcal{R}}$, the operators L^0 and L^1 are defined in Equation (21).

$$\begin{aligned} L^0 &\equiv f(X(t)) \frac{\partial}{\partial X} + \frac{1}{2}\sigma^2(X(t)) \frac{\partial^2}{\partial X^2}, \\ L^1 &\equiv \sigma(X(t)) \frac{\partial}{\partial X}. \end{aligned} \quad (21)$$

3 Methodologies

Several implementations of epidemic models using stochasticity to varying degrees are explored in this section by adding more complex elements each time with the aim of better capturing an epidemic projection.

3.1 Chain Binomial model (Reed-Frost)

The Reed-Frost model ([Frost, 1928](#); [De Oliveira Costa Maia, 1952](#)) is a simple, discrete time stochastic model known as a 'chain binomial'. The model is best suited to closed population such as households, schools, or hospitals. The following derivation of the model adapts both the original work, as well as a more succinct explanation by [Bailey \(1964\)](#).

Using the SIR model as an example, the compartments can be split as follows:

S_t - susceptible individuals at time t .

I_t - Infectious individuals at time t (also define I_t as a random variable).

R_t - Recovered individuals at time t .

The probability of an infection from an exposed individual can also be defined as p , whilst the total population size is $N = S + I + R$. A set of assumptions are then made for the model, namely,

- (i) The population is fixed and well-mixed with no births or deaths occurring.
- (ii) The disease is transferred from one infectious individual to another whenever contact is made.
- (iii) Contacts infectious people make are independent of each other and the probability of a susceptible person becoming infected is proportional to the number of infectious people.

From here, two directions can be taken: deterministic updates of infectious individuals derived from the expectation, or stochastic updates with Bernoulli trials. Here, the latter is explored, however the deterministic form can be found at the beginning of a paper by [Fine \(1977\)](#). Both versions share identical expressions for updates of susceptible and recovered individuals and these updates, at an incremented time $t + 1$, are formulated in the following Equations [\(22\)](#) and [\(23\)](#),

$$S_{t+1} = S_t - I_{t+1}, \quad (22)$$

$$R_{t+1} = R_t + I_t. \quad (23)$$

For updates regarding the number of infected individuals at time $t + 1$, the probability of getting infected by any infectious individual needs to be calculated. First, consider the probability to *not* get infected by a particular individual, $(1 - p)$, then the probability to not become infected by any infected individual is done by

raising to the power of the total infectious individuals i.e. $(1 - p)^{I_t}$. The probability of getting infected by any infectious individual is subsequently expressed by Equation (24),

$$1 - (1 - p)^{I_t}. \quad (24)$$

For simplicity, let $q = 1 - p$, which now gives the probability as

$$1 - q^{I_t}. \quad (25)$$

The probability of no infection can now be simply stated in Equation (26),

$$1 - (1 - q^{I_t}) = q^{I_t}. \quad (26)$$

Next, recall the Binomial distribution (Def 2.5),

$$\mathbb{P}[X = x] = \binom{n}{x} p^x (1 - p)^{n-x}, \quad (27)$$

the probability of infecting, say x , new individuals can be modelled on this binomial distribution, with the random variable $I_{t+1} \sim \text{Bin}(S_t, 1 - q^{I_t})$. This results in the following probability for infecting a certain number of new individuals derived in Equation (28):

$$\mathbb{P}[I_{t+1} = x] = \binom{S_t}{x} (1 - q^{I_t})^x (q^{I_t})^{S_t - x}. \quad (28)$$

Infection numbers are updated through simulation of Bernoulli trials and the compartments in Equation (22) and (23) can also be subsequently updated. The basic reproductive number for this model is expressed as $\mathcal{R}_0 = (N - 1)(1 - q)$, and from this, the probability q needed for the model can be derived. A numerical simulation of the Reed-Frost model can be found in the Results section (Figure 5).

A similar model based on chain binomials was proposed by [Greenwood \(1931\)](#), and differs in that it assumes the chance of infection is independent from the current number of infectious individuals. Another interesting property of the chain binomial model is the absence of dependence on earlier time steps; with the model only relying on the previous generation to inform the next size of each compartment. This is analogous to the Markov property, thus chain binomials are examples of (bivariate) Markov chains, with researches such as [Gani and Jerwood \(1971\)](#) defining as such. The chain binomial model derived from Reed-Frost is not necessarily the only direction, with work by [Frauenthal \(1980\)](#) and [Becker \(1981\)](#) applying these methods onto smaller samples (households for example), ‘chain’ together the different outcomes, and then state the probability to which they occur.

3.2 Branching Processes

In the introductory section, a branching process was suggested as a possible stochastic avenue that can be taken to model early stage epidemic processes. The method can also be used to predict an end to the spread of a disease, more specifically to calculate the probability of extinction. The branching process traces its ancestry back to work by mathematician Francis Galton and his study into the extinction of family names ([Kendall, 1966](#)). Put forward as problem 4001 in the academic journal *Educational Times* in 1873, it stated the following:

“Problem 4001. A large nation, of whom we will only concern ourselves with the adult males, N in number, and who each bear separate surnames, colonise a district. Their law of population is such that, in each generation, a_0 per cent of the adult males have no male children who reach adult life; a_1 have one such male child; a_2 have two; and so on up to a_5 , who have five. Find (1) what proportion of the surnames will have become extinct after r generations; and (2) how many instances there will be of the same surname being held by m persons.”

Unable to initially find satisfactory answers, friend and fellow mathematician Rev. Henry Watson submitted a solution a few months after publication, which resulted in a joint presentation by Galton and Watson in 1874 ([Albertsen, 1995](#)). Despite this publication not fully answering the question at hand, with early 20th century mathematicians Fisher, Haldane, Erlang, and Steffenson providing more complete results ([Allen, 2010](#)), the branching problem is still dedicated and named after the two initial contributors as the Galton-Watson process. Alternatively, the Bienaym  -Galton-Watson process is used to describe the method, due to the independent study by French mathematician Ir  n  e-Jules Bienaym  , which concerns a similar problem to that put forward by Galton ([Albertsen, 1995](#)). In an epidemiological setting, the problem can be transformed from that concerning the persistence of a family’s surname, to the persistence of a disease between contacts. Male children reaching adult life are replaced with susceptible individuals and the population laws for generations can be modelled using probability generating function, to serve as a tool for the number of potential future infections. [Allen \(2017\)](#) describes such framework in epidemiology as “the linear approximation ... near the disease-free equilibrium”.

Before proceeding with precise definitions, two important assumptions for this particular branching process, consistent with that of all Galton-Watson processes, need to be made, namely:

- (i) Every infectious individual is independent from each other.
- (ii) the susceptible population is sufficiently large.

Focusing solely on the spread of infectious individuals, let I_t denote the number of new cases at a time t . In the next time step, $t + 1$, new infectious individuals infect on average \mathcal{R}_0 susceptibles. If p_j denoted the probabilities that a single infectious

individual infects j individual, then the corresponding pgf for the number of new infections is given by the following Equation (29),

$$G(t) = \sum_{j=0}^{\infty} p_j t^j. \quad (29)$$

The expectation is given as the first derivative, evaluated at $t = 1$ i.e. $G'(1) = \mathcal{R}_0$.

Perhaps the most unique and important result that can be derived from the Bienaymé-Galton-Watson is the probability of the epidemics extinction, expressed as the following limit in Equation (30),

$$\lim_{t \rightarrow \infty} \mathbb{P}[I_t = 0]. \quad (30)$$

This probability depends on $G(t)$ and is attached to the following corollary, stated by [Allen \(2008\)](#).

Corollary 3.1 (Probability of Extinction). *If the pgf $G(t)$ satisfies $0 \leq p_0 + p_1 \leq 1$ and there is an arbitrary number of initial infectious individuals, $i_0 > 0$, such that $\mathbb{P}[I_t = i_0] = 1$, then,*

1. *If $\mathcal{R}_0 \leq 1$, then extinction is guaranteed ($\lim_{t \rightarrow \infty} \mathbb{P}[I_t = i_0] = 1$).*
2. *If $\mathcal{R}_0 > 1$, then the probability of extinction is given as:*

$$\lim_{t \rightarrow \infty} \mathbb{P}[I_t = i_0] = q^{i_0},$$

where q is a unique fixed point in $[0, 1]$ such that $G(q) = q$.

3. *The probability of a disease becoming endemic in the population, given $\mathcal{R}_0 > 1$, is $1 - q^{i_0}$.*

This corollary is an extension of a general theorem concerning the general extinction of a branching process ([Allen, 2010](#), p. 166-170).

Example 3.1. Suppose there is a disease in which the number of cases follow a Poisson distribution, $I_t \sim \text{Pois}(\mathcal{R}_0)$ ([Antia et al., 2003](#)). Then, the corresponding pgf is calculated as:

$$\begin{aligned} G(t) &= \sum_{j=0}^{\infty} e^{-\mathcal{R}_0} \frac{\mathcal{R}_0^j}{j!} t^j \\ &= e^{-\mathcal{R}_0} \sum_{j=0}^{\infty} \frac{(\mathcal{R}_0 t)^j}{j!} \\ &= e^{-\mathcal{R}_0} e^{\mathcal{R}_0 q} \\ &= e^{-\mathcal{R}_0(1-q)}, \end{aligned} \quad (31)$$

where the Taylor series expansion of the exponential function ($e^x = \sum_n x^n / n!$) is used. If this disease has initial exponential growth with $\mathcal{R}_0 = 1.2$, then by applying

Corollary 3.1, the probability the disease becomes persistent and endemic in the population is found by first solving the fixed point (see Equation 32),

$$q = e^{-1.2(1-q)}. \quad (32)$$

Figure 3 gives the graphic representation alongside the intersections, which are solved numerically.

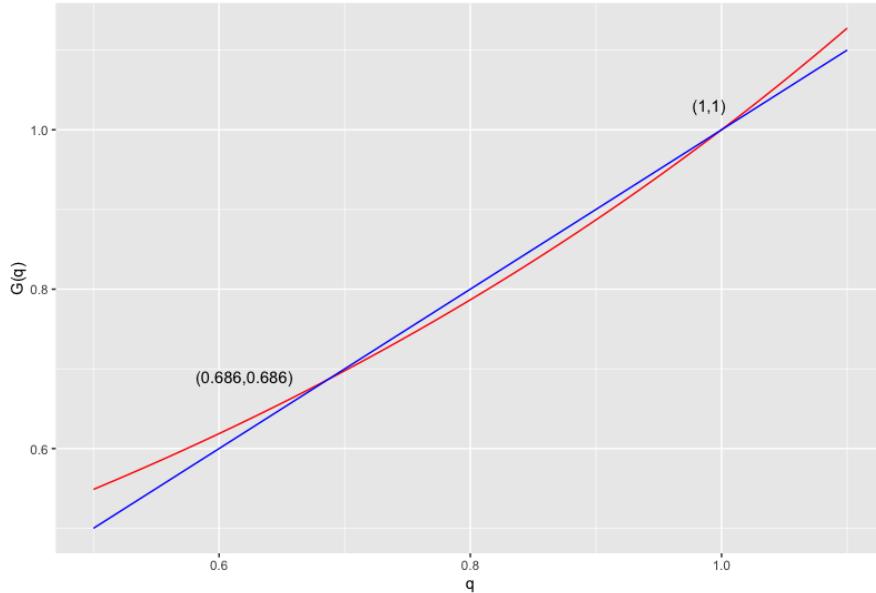


Figure 3: The graph of the intersection point of $G(q)=q$. A trivial point exists at $q=1$, and the non-trivial intersection at $q = 0.686$. The function $G(q)$ is in red, and the straight line q in blue.

From this, with an initial number of infections $i_0 = 4$ i.e. $\mathbb{P}[I_t = 4] = 1$, the probability of an endemic disease is simply,

$$1 - q^{i_0} = 1 - 0.686^4 \approx 0.779. \quad (33)$$

The branching process can also be extended to the SIR model (Allen, 2017).

3.3 Stochastic Differential Equations

The origins of stochastic differential equations are derived from work by career physicists Einstein and Smoluchowski, using the random phenomenon of Brownian motion to derive partial differential equations for brownian particle displacement ([Sobczyk, 1991](#)). In the context to be used for a general stochastic process, this came in work by Japanese mathematician Kiyosi Itô and the formulation of the Itô integral (see section 3.3). SDEs differ from their ODE counterparts with the addition of a random process based around Brownian motion, or in this case, Wiener processes. While not primarily derived for use in epidemiology, rather for financial mathematics, there is still much applicability to the field and provides a stable, more complex alternative to chain binomials.

3.3.1 Derivation

Recall the deterministic SIR model without vital dynamics (Sec. 2.1.1) and redefine $S(t), I(t)$ as random variables (there is no need to define $R(t)$ as such since it can be recovered from $R(t) = N - S(t) - I(t)$) over a continuous interval:

$$S(t), I(t) \in [0, N].$$

Then, the change of system variables are expressed by the vector $\Delta X(t)$ in the following Equation (34),

$$\Delta X(t) = [\Delta S(t) \ \Delta I(t)]^T, \quad (34)$$

where t is subdivided into small sub-intervals, Δt , which can then be further subdivided as shown in Equation (36),

$$\Delta t_i = t_i - t_{i-1}, i = 1, \dots, n \quad (35)$$

$$\sum_{i=1}^n \Delta t_i = \Delta t. \quad (36)$$

Note, $t_0 = t$ and $t_n = t + \Delta t$. This results in $\Delta X(t)$ being re-expressed in Equation (37):

$$\Delta X(t) = \sum_{i=1}^n \Delta X(t_i). \quad (37)$$

Given Δt_i is sufficiently small, the family of random variables $\{\Delta X(t_i)\}_{[0,t]}$ on the interval Δt are i.i.d. Then for a sufficiently large n , by the Multivariate Central Limit Theorem (see Theorem 2.4), $\Delta X(t)$ has an approximate normal distribution with mean $\mathbb{E}[\Delta X(t)]$ and co-variance matrix $\text{cov}[\Delta X(t)]$. Thus,

$$\Delta X(t) - \mathbb{E}[\Delta X(t)] \approx \text{Normal}(\mathbf{0}, \text{cov}[\Delta X(t)]),$$

where $\mathbf{0}$ is the zero vector. The mean of ΔX is simply the deterministic version of $S(t)$ and $I(t)$ to order Δt :

$$\mathbb{E}[\Delta X] \approx [-\beta SI/N \ \beta SI/N - \gamma I]^T \Delta t = f(X(t)) \Delta t. \quad (38)$$

The co-variance matrix is expressed as,

$$\begin{aligned}
 \text{cov}[\Delta X] &\approx \mathbb{E}[\Delta X(\Delta X)^T] \quad (\text{by definition of co-variance}) \\
 &= \mathbb{E} \left[\begin{pmatrix} (\Delta S)^2 & \Delta S \Delta I \\ \Delta S \Delta I & (\Delta I)^2 \end{pmatrix} \right] \\
 &= \begin{pmatrix} \beta SI/N & \beta SI/N \\ -\beta SI/N & \beta SI/N + \gamma I \end{pmatrix} \Delta t \\
 &= \mathbf{C} \Delta t.
 \end{aligned} \tag{39}$$

It is now left to compute a matrix \mathbf{G} such that $\mathbf{G}\mathbf{G}^T = C$. This resulting matrix is expressed as:

$$\mathbf{G} = \begin{pmatrix} -\sqrt{\beta SI/N} & 0 \\ \sqrt{\beta SI/N} & -\sqrt{\gamma I} \end{pmatrix} \tag{40}$$

(see [Allen \(2017\)](#) for computation to find \mathbf{G}).

Finally, $\Delta X \approx f(X(t)) + \mathbf{G}\Delta W(t)$, where $W(t) = [\Delta W_1(t) \ \Delta W_2(t)]^T$ and $\Delta W_i(t) \sim \text{Normal}(0, \Delta t)$. Letting $\Delta t \rightarrow 0$ results in the following Equation (41),

$$dX(t) = f(X(t))dt + \mathbf{G}dW(t). \tag{41}$$

It can be seen that this takes the form of an Itô SDE (Def. [2.12](#)), with two independent Wiener processes $W(t) = [W_1(t) \ W_2(t)]^T$. Substituting in the values for vector $X(t)$ and matrix \mathbf{G} , a set of SDEs for our susceptible and infected groups is formalised.

$$\begin{aligned}
 dS(t) &= - \left[\frac{\beta S(t)I(t)}{N} \right] dt - \sqrt{\frac{\beta S(t)I(t)}{N}} dW_1(t), \\
 dI(t) &= \left[\frac{\beta S(t)I(t)}{N} - \gamma I(t) \right] dt + \sqrt{\frac{\beta S(t)I(t)}{N}} dW_1(t) - \sqrt{\gamma I(t)} dW_2(t).
 \end{aligned} \tag{42}$$

In this model, the epidemic stops when $I(t) = 0$, and neglecting the Wiener process will clearly lead back to our deterministic formulation in Equation (5). For the SDE model including vital dynamics see [Allen \(2008\)](#), and for real life application to an infectious disease, see [Maki and Hirose \(2013\)](#).

3.4 Simulation of Epidemic Models

To simulate stochastic models, a numerical method or stochastic simulation algorithm is often employed to calculate various sample paths. These methods have largely been created outside of the field of Epidemiology, but can still equally be applied to an epidemic model such as SIR, and require minimal set up for implementation. For continuous time models, often with an underlying Markov Chain, the Gillespie stochastic simulation algorithm is an appropriate choice. For stochastic differential equations, a family of methods is explored with preference as to which best fits the scenario stated.

3.4.1 Gillespie's Algorithm

Stemming from the work of [Gillespie \(1976, 1977\)](#) on chemical reactions using stochastic methods to simulate the time evolution of a chemical system, Gillespie's Stochastic Simulation Algorithm (sometimes denoted SSA in literature) can also be modelled to an epidemiological context. In general, it is a method to simulate potential trajectories of finite populations in a continuous time environment.

Before introducing the algorithm, a few key parameters and variables are to be touched upon, with full definitions of all the following variables for Gillespie's Algorithm found in [Appendix A](#). The state vector $\mathbf{X}(t)$, records the systems configurations, with:

$$\mathbf{X}(0) = (X_1(0), X_2(0), X_3(0), \dots, X_n(0)), \quad (43)$$

giving the initial state. Then, the probabilities at which reactions occur over an infinitesimal interval $[t + \tau, t + \tau + dt]$, is expressed as the propensity vector \mathbf{a} . Note, τ is the sojourn time till the next reaction. Lastly, the state change - or stoichiometry - matrix \mathbf{v} , denotes the state changes between the reactions. The state vector updates to $\mathbf{X}(t + \tau)$, in accordance to a monte carlo method which selects from the stoichiometry vector an update for each state. This is iterated until either a final time or final reaction criterion is reached. [Algorithm 1](#) runs through the Direct method of Gillespie, which is the most common and referred to variation, albeit is not the primary choice for computational simulation (this is touched upon in the results [Section 5](#)).

Algorithm 1: Gillespie's Direct Algorithm

```

input:  $\mathbf{X}_0, \mathbf{a}, \mathbf{v} t_{\text{final}}, r_{\text{total}}$ 
 $r_1, r_2 \leftarrow$  Uniform Random Numbers
 $t, n \leftarrow 0$ 
 $\mathbf{X} \leftarrow \mathbf{X}_0$ 
while ( $t < t_{\text{final}}$  OR  $n < r_{\text{total}}$ ) do
     $a_0 \leftarrow \sum_j a_j$ 
     $\tau \leftarrow \frac{1}{a_0} \log(r_1)$ 
     $j' \leftarrow \text{SELECT } j' \text{ s.t. } \sum_{j=1}^{j'-1} a_j < r_2 a_0 \leq \sum_{j=1}^{j'} a_j$ 
     $t \leftarrow t + \tau$ 
     $\mathbf{X} \leftarrow \mathbf{X} + v_{j'}$ 
end

```

Example 3.2 (SIR model). Consider the SIR model without vital dynamics ([Sec. 2.1.1](#)). The system is made up of two reaction expressed in [Equation \(44\)](#), which then corresponds to propensity functions $a_1 = \beta SI/N$ and $a_2 = \gamma I$.



Given there are 3 states (Susceptible, Infected, and Recovered) and 2 reactions, the stoichiometry matrix is given as the 3×2 matrix, \mathbf{v} . Here, v_{ij} corresponds to the change in the number of individuals in state i after reaction j ; defined by the aforementioned propensity functions a_j . Thus,

$$\mathbf{v} = \begin{pmatrix} -1 & 0 \\ +1 & -1 \\ 0 & +1 \end{pmatrix}. \quad (45)$$

The initial state vector corresponds to the initial values of each state, with,

$$\mathbf{X}(0) = [S(0) \ I(0) \ R(0)]^T. \quad (46)$$

Like other initial conditions for this SIR model, given an initial fixed population size N , it is common to set $S(0) = N - 1$, $I(0) = 1$, $R(0) = 0$. The probabilities of each event occurring is fairly straightforward in this case, as only two reactions exist. These probabilities are given by Equations (47) and (48).

$$p_{\text{infection}} = \frac{\beta SI/N}{\beta SI/N + \gamma I} \quad (47)$$

$$p_{\text{recovery}} = \frac{\gamma I}{\beta SI/N + \gamma I}. \quad (48)$$

The algorithm can then be run with a halting condition of either the number of days to monitor or when the epidemic dies out. A numerical simulation of Gillespie's Algorithm, using the computational preference of an optimised tau-leaping method as opposed to the direct method, can be found in Section 5.1. The differences between the two algorithms are minimal with the key difference being in how the time steps are calculated. Tau-leaping often takes larger time steps in order to speed up the computational process.

3.4.2 Numerical methods for SDEs

Relating to matters on the solution of SDEs, explicit solutions are rarely achievable. Instead, numerical methods are employed to approximate solutions. In the field of SDEs, many exist and are often extensions of those used to solve ordinary differential equations. Euler-Maruyama and stochastic Runge-Kutta ([Wilkie, 2004](#)) are adaptations of their ODE counterparts, whilst original methods have also been proposed, a popular example being [Milstein \(1975\)](#). For brevity, only the Euler-Maruyama and Milstein's methods will be touched upon as these are the most prominent methods offered in software libraries for simulation of SDEs. For examples of this see [Guidoum and Boukhetala \(2020\)](#) and the companion package to the book by [Iacus \(2008\)](#); the latter also serves as partial reference for the derivation of the following methods.

Euler-Maruyama

Perhaps the simplest way to approximate a SDE is the Euler-Maruyama method. Like its non-stochastic counterpart, Euler's method, it can be derived as the first order Taylor series Expansion of the Itô SDE (Def. 2.13). The recursive definition can be seen in Equation (49), where the interval $[0, T]$ is partitioned into N sub-intervals, each of width Δt ,

$$X(t_{i+1}) = X(t_i) + f(X(t_i))\Delta t + \sigma(X(t_i))\Delta W_i, \quad (49)$$

where $\Delta t = t_{i+1} - t_i$ and $\Delta W_i = W(t_{i+1}) - W(t_i)$ for $i = 0, 1, \dots, N - 1$. An initial value is also set as $X(t_0) = X(0)$.

In the case of the dynamics described previously in Equation (42), of the SIR stochastic differential equation, a scheme for the iterative method can be constructed as,

$$S(t_{i+1}) = S(t_i) + \left[-\frac{\beta S(t_i)I(t_i)}{N} \right] \Delta t + \left[\sqrt{\frac{\beta S(t_i)I(t_i)}{N}} \right] \Delta W_i, \quad (50)$$

$$\begin{aligned} I(t_{i+1}) &= I(t_i) + \left[\frac{\beta S(t_i)I(t_i)}{N} - \gamma I(t_i) \right] \Delta t \\ &\quad + \left[\sqrt{\frac{\beta S(t_i)I(t_i)}{N}} - \sqrt{\gamma I(t_i)} \right] \Delta W_i, \end{aligned} \quad (51)$$

$$R(t_{i+1}) = N - S(t_{i+1}) - I(t_{i+1}), \quad (52)$$

with the Wiener process given as $\Delta W_i \sim \text{Normal}(0, \Delta t)$.

Milstein method

The method here is one of two proposed in work by Milstein (1975). It can be simply derived by taking the Taylor Series approximation to second order terms i.e.

$$\begin{aligned} X(t_{i+1}) &= X(t_i) + f(X(t_i))\Delta t + \sigma(X(t_i))\Delta W_i \\ &\quad + \frac{1}{2}\sigma(X(t_i))\sigma'_x(X(t_i))[(\Delta W_i)^2 - \Delta t]. \end{aligned} \quad (53)$$

Again, $i = 0, 1, 2, \dots, N - 1$, $X(t_0) = X_0$, and in addition, The function $\sigma'_x(x)$ is the partial derivative of $\sigma(x)$ with respect to x . If this function was determined to be plain additive noise (constant), then the approximation reduces to the Euler-Maruyama scheme. In the case of the SIR stochastic differential equation, described in Section 3.3, scaled additive noise was introduced therefore this method is applicable in running stochastic realisations. A drawback of Milstein, especially in the case of SDEs with two variables and the requirement of multi-dimensional Wiener processes, is the generation of said Wiener processes. The issue is raised in a paper by Alnafisah (2018), with subsequent solution in the form of Fourier series expansion. Since the Euler-Maruyama scheme is already

an effective numerical simulations of the SIR stochastic differential equation, this simulation is focused on and a computational simulation is found in Section 5.1. This is also the preferred method for later models using SDEs.

3.4.3 Convergence approximations of numerical SDE methods

An important concept for numerical methods to approximate SDEs, are the orders of convergence. These fall into two categories: weak and strong orders of convergence, used to evaluate the optimality of numerical methods ([Iacus, 2008](#)). They are given by the following definitions.

Definition 3.1 (Strong order of Convergence). Consider an approximation, Y_δ , of a continuous time process Y , where δ is the maximum time between approximations. This is said to hold strong order of convergence γ to the approximation Y if, for a fixed time T in the future, it holds that

$$\mathbb{E} [|Y_\delta(T) - Y(T)|] \leq C\delta^\gamma, \quad \forall \delta < \delta_0, \quad (54)$$

for $\delta_0 > 0$ and constant C .

A similar definition can also be defined for Weak convergence.

Definition 3.2 (Weak order of Convergence). Y_δ converges to Y of weak order β if, for a fixed future time T and any $2(\beta + 1)$ continuously differentiable function g , it holds that

$$|\mathbb{E} [g(Y(T))] - \mathbb{E} [g(Y_\delta(T))]| \leq C\delta^\beta, \quad \forall \delta < \delta_0, \quad (55)$$

for $\delta_0 > 0$ and constant C .

In informal terms, it can be seen that weak convergence is the measure of the error of the mean, whilst strong convergence is essentially the mean of the error. For weak order, an intuitive explanation of how the order is found is to start with the expected value of the function $g(Y(t))$, which is estimated through Monte Carlo simulations. Then, by plotting the left hand side of Equation (55) against different time discretizations and fitting a least square fit, the order is attained. Calculation of strong order follows a similar path. For the methods mentioned in this thesis, Euler-Maruyama has strong order $\gamma = 1/2$, and Milstein has superior strong and weak orders of $\beta = \gamma = 1$ ([Iacus, 2008](#)).

4 A stochastic model for the spread of malaria

The following model is adapted from a number of deterministic and stochastic models in contemporary and historic literature, namely those by [Allen \(2017\)](#); [Macdonald \(1956\)](#); [Tumwiine *et al.* \(2007\)](#); [Wedajo *et al.* \(2018\)](#); [Witbooi *et al.* \(2020\)](#). By beginning with some underlying assumptions and creating the compartmental structure, a deterministic model can be formulated. From this, a stochastic model can then be derived.

The underlying compartmental model is a variation on a SIRS-SI model, without vital dynamics for the human population, but with vital dynamics for the vector (in this case mosquito population). This latter condition is a consequence of the short life span of mosquitoes, with life-cycle estimates ranging from a few days to a few months ([Matthews *et al.*, 2020](#)). Assuming a relatively short span of time and a healthy population, neglecting vital dynamics in humans is a reasonable assumption to make for model simplification; this is not the case for mosquito populations and inclusion of parameters for births and deaths are a necessity. However, it can still be reasonable to assume a stable, constant population within the mosquito community. This is achieved by setting the birth rate equal to the death rate, and the parameter μ is used to denote as such. The recovery rate of a human, once infected, is given by γ . For the constant human and mosquito populations, H and M represents the populations respectively. Focusing on the human compartments, a modification of the basic SIR model discussed thus far is needed to adapt to the lack of people developing permanent immunity to malaria. Although there is no definitive answer within the research community on how malaria immunity works ([Doolan *et al.*, 2009](#)), reinfection is likely if exposed again. Therefore, once a person moves into the recovery stage, after a certain period of time, they would move back to the susceptible population. A feature incorporated into a model by [Aron and May \(1982\)](#), it is addressed with the addition of a parameter to denote the per capita loss of temporary immunity (θ).

Since malaria cannot pass from human to human, transmission parameters are introduced. Two scenarios are considered in transmission, this being from an infected mosquito into a susceptible human host (α_1) or a mosquito acquiring malaria from the blood of an infected human (α_2). These rates are proportionate to the rate of bites a mosquito gives to fulfil its blood requirement coupled with the probability of infection of a bite from either human to mosquito or vice versa.

To summarise, the list of the above mentioned assumptions, among others, can now be given in a formal manner:

- (i) The human and mosquito population is fixed with vital dynamics for mosquito population only.
- (ii) Malaria infection starts at the point of an infected *female* mosquito bites a susceptible human (from henceforth all mention of a mosquito population refers only to females).
- (iii) Once recovered, humans have temporary immunity before being reintroduced to the susceptible population.

- (iv) All mosquitoes bite humans in the susceptible and infected population at random, with transmission from vector to human, or human to vector both possible. Although infected mosquitoes biting infected humans are ignored.
- (v) Mosquitoes, once infected, cannot recover and eventually die off.

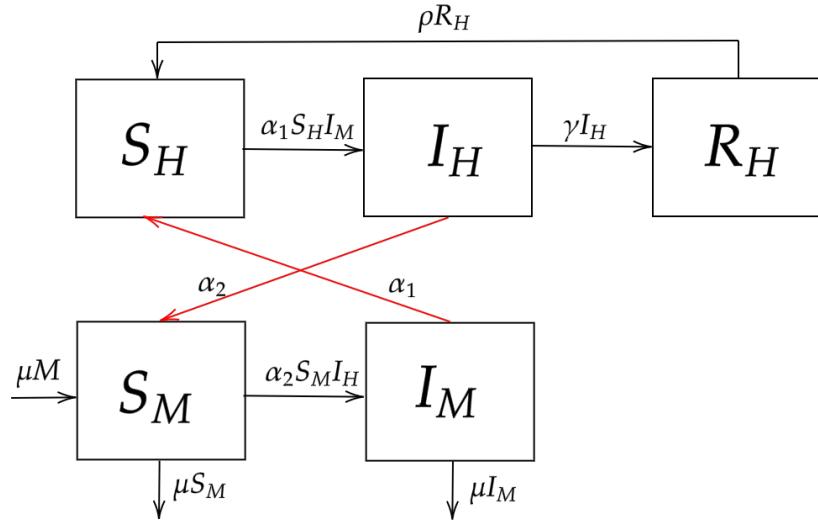


Figure 4: A schematic diagram for the compartmental model of the spread of malaria in a population. The human population (H) remains constant, with births and deaths neglected. The mosquitoes (M) have constant population, but births and deaths are not neglected.

To aid with visualisation of the compartments and transitions, a schematic diagram for this model is presented in Figure 4. A formal list of the parameters and constants used in current, prior, and subsequent models, is also provided and resides in the Nomenclature.

4.1 Deterministic model and properties

A deterministic formulation of the model, based on the schematic diagram, is given by the system of ODEs in Equations (56) to (60) with conditions on population given by Equation (61) and (62).

$$\frac{dS_H}{dt} = -\alpha_1 \frac{S_H}{H} I_M + \theta R_H, \quad (56)$$

$$\frac{dI_H}{dt} = \alpha_1 \frac{S_H}{H} I_M - \gamma I_H, \quad (57)$$

$$\frac{dR_H}{dt} = \gamma I_H - \theta R_H, \quad (58)$$

$$\frac{dS_M}{dt} = -\alpha_2 S_M \frac{I_H}{H} + \mu M - \mu S_M, \quad (59)$$

$$\frac{dI_M}{dt} = \alpha_2 S_M \frac{I_H}{H} - \mu I_M, \quad (60)$$

$$H = S_H + I_H + R_H, \quad (61)$$

$$M = S_M + I_M. \quad (62)$$

All parameters are assumed strictly positive and since R_H can be expressed by $R_H = H - S_H - I_H$, the above system may be reduced to 4 equations if desired.

4.1.1 Disease-free equilibrium

If there is no malaria in the human population and parasite carrying mosquitoes are absent in the vector population, the system is said to be in a disease-free equilibrium. The conditions to recover what the values the variables need to attain to reach this equilibrium is recovered from setting each ODE in the Equations (56) to (60) equal to 0 i.e.

$$-\alpha_1 \frac{S_H}{H} I_M + \theta R_H = 0, \quad (63)$$

$$\alpha_1 \frac{S_H}{H} I_M - \gamma I_H = 0, \quad (64)$$

$$\gamma I_H - \theta R_H = 0, \quad (65)$$

$$-\alpha_2 S_M \frac{I_H}{H} + \mu M - \mu S_M = 0, \quad (66)$$

$$\alpha_2 S_M \frac{I_H}{H} - \mu I_M = 0. \quad (67)$$

By the definition of what is needed to attain this equilibrium, $I_h = I_M = 0$. Given this, it can be seen in Equation (65) that $R_H = 0$. From Equation (61), $S_H = H$ and likewise from Equation (62), $S_M = M$. Therefore the disease-free equilibrium is denoted in the following Equation (68):

$$(\bar{S}_H, \bar{I}_H, \bar{R}_H, \bar{S}_M, \bar{I}_M) = (H, 0, 0, M, 0). \quad (68)$$

4.1.2 Basic reproduction number

For this system, the \mathcal{R}_0 value cannot just be expressed as the ratio between infection and recovery parameters, which was the case in the SIR model without vital dynamics. As noted by [van den Driessche and Watmough \(2002\)](#), this heuristic definition of the basic reproduction rate is simply not suffice, rather a technique known as a next generation matrix is preferred (see Section [2.1.2](#) for brief overview of the method). In the SIRS-SI compartmental model used thus far, the system can be grouped into disease classes ($dI_H/dt, dI_M/dt$) and non disease classes ($dS_H/dt, dS_M/dt, dR_H/dt$). The vector \mathcal{F} contains terms with a secondary infection rate in the disease classes, and \mathcal{V} , the remaining terms in the disease classes. This yields the following in Equations [\(69\)](#) and [\(70\)](#),

$$\mathcal{F} = \begin{bmatrix} \alpha_1 \frac{S_H}{H} I_M \\ \alpha_2 S_M \frac{I_H}{H} \end{bmatrix} \quad (69)$$

$$\mathcal{V} = \begin{bmatrix} \gamma I_H \\ \mu I_M \end{bmatrix}. \quad (70)$$

By letting functions,

$$\begin{aligned} g_1 &= \alpha_1 \frac{S_H}{H} I_M, \\ g_2 &= \alpha_2 S_M \frac{I_H}{H}, \\ h_1 &= \gamma I_H, \\ h_2 &= \mu I_M, \end{aligned}$$

the matrices \mathbf{F} and \mathbf{V} are then given by Equation [\(71\)](#).

$$\begin{aligned} \mathbf{F} &= \begin{pmatrix} \frac{\partial g_1}{\partial I_H} & \frac{\partial g_1}{\partial I_M} \\ \frac{\partial g_2}{\partial I_H} & \frac{\partial g_2}{\partial I_M} \end{pmatrix} = \begin{pmatrix} 0 & \alpha_1 \frac{S_H}{H} \\ \alpha_2 \frac{S_M}{H} & 0 \end{pmatrix}, \\ \mathbf{V} &= \begin{pmatrix} \frac{\partial h_1}{\partial I_H} & \frac{\partial h_1}{\partial I_M} \\ \frac{\partial h_2}{\partial I_H} & \frac{\partial h_2}{\partial I_M} \end{pmatrix} = \begin{pmatrix} \gamma & 0 \\ 0 & \mu \end{pmatrix}. \end{aligned} \quad (71)$$

The spectral radius $\rho(\mathbf{F}\mathbf{V}^{-1})$ (see Def. 2.2) is computed in Equation (72).

$$\begin{aligned}\rho(\mathbf{F}\mathbf{V}^{-1}) &= \rho\left(\begin{pmatrix} 0 & \alpha_1 \frac{S_H}{H} \\ \alpha_2 \frac{S_M}{H} & 0 \end{pmatrix} \begin{pmatrix} \gamma^{-1} & 0 \\ 0 & \mu^{-1} \end{pmatrix}\right) \\ &= \rho\left(\begin{pmatrix} 0 & \alpha_1 \mu^{-1} \frac{S_H}{H} \\ \alpha_2 \gamma^{-1} \frac{S_M}{H} & 0 \end{pmatrix}\right) \\ &= \sqrt{\frac{\alpha_1 \alpha_2 S_H S_M}{H^2 \gamma \mu}}.\end{aligned}\tag{72}$$

Evaluated at the disease-free equilibrium ($S_H = \bar{S}_H$ and $S_M = \bar{S}_M$), this becomes,

$$\rho(\mathbf{F}\mathbf{V}^{-1}) = \sqrt{\frac{\alpha_1 \alpha_2 M}{H \gamma \mu}}.\tag{73}$$

The next step then depends on preferences surrounding whether to set $\mathcal{R}_0 = \rho(FV^{-1})$ or its square. Regardless of the choice, the behaviour of $\mathcal{R}_0 \leq 1$ remains the same, the preference in this thesis is to take \mathcal{R}_0 as $\rho(FV^{-1})^2$ or,

$$\mathcal{R}_0 = \left(\frac{M}{H}\right) \left(\frac{\alpha_1 \alpha_2}{\gamma \mu}\right).\tag{74}$$

The disease dies out if $\mathcal{R}_0 < 1$ and a stable endemic equilibrium exists if $\mathcal{R}_0 > 1$ (Allen, 2017).

4.2 Stochastic vector-borne model

A stochastic compartmental model can be constructed from the schematic diagram in Figure 4. The choice remains as to whether to construct the stochastic model as either a continuous time Markov chain, in the same vein of one used for the purpose of carrying out Gillespie's algorithm, or as a system of Itô SDEs, and subsequently evaluated with a numerical method. The latter is prioritised and is based on an underlying deterministic model given in Section 4.1.

The model broadly follows along with example used in Section 3.3 for a SIR model without vital dynamics. In this case, there are five random variables and each can be stored in a change of system variable vector $\Delta X(t)$:

$$S_H(t), I_H(t), R_H(t), S_M(t), I_M(t) \in [0, N],\tag{75}$$

$$\Delta X(t) = \begin{bmatrix} \Delta S_H(t) & \Delta I_H(t) & \Delta R_H(t) & \Delta S_M(t) & \Delta I_M(t) \end{bmatrix}^T.\tag{76}$$

The goal is to create a system of SDEs of the form of the Itô SDE,

$$dX(t) = f(X(t))dt + \mathbf{G}dW(t).\tag{77}$$

Recall that the matrix \mathbf{G} is interchangeable with the function $\sigma(X(t))$. From Equation (38), $f(X(t))\Delta t \approx \mathbb{E}[\Delta X]$ and, taking $\Delta t \rightarrow 0$, gives $f(X(t))dt$. More specifically this equates to the following in Equation (78),

$$f(X(t))dt = \begin{bmatrix} -\alpha_1 \frac{S_H}{H} I_M + \theta R_H \\ \alpha_1 \frac{S_H}{H} I_M - \gamma I_H \\ \gamma I_H - \theta R_H \\ -\alpha_2 S_M \frac{I_H}{H} + \mu M - \mu S_M \\ \alpha_2 S_M \frac{I_H}{H} - \mu I_M \end{bmatrix}. \quad (78)$$

The matrix \mathbf{G} can be found from $\mathbf{G}\mathbf{G}^T = \mathbf{C}$, where \mathbf{C} is the co-variance matrix, formulated as follows and given by Equation (80).

$$\begin{aligned} \text{cov}[\Delta X] &\approx \mathbb{E}[\Delta X(\Delta X)^T] \\ &= \mathbf{C}\Delta t \end{aligned} \quad (79)$$

$$= \left(\begin{array}{ccccc} \alpha_1 \frac{S_H}{H} I_M + \theta R_H & -\alpha_1 \frac{S_H}{H} I_M & -\theta R_H & 0 & 0 \\ -\alpha_1 \frac{S_H}{H} I_M & \alpha_1 \frac{S_H}{H} I_M + \gamma I_H & -\gamma I_H & 0 & 0 \\ -\theta R_H & -\gamma I_H & \theta R_H + \gamma I_H & 0 & 0 \\ 0 & 0 & 0 & \alpha_2 S_M \frac{I_H}{H} + \mu M + \mu S_M & -\alpha_2 S_M \frac{I_H}{H} \\ 0 & 0 & 0 & -\alpha_2 S_M \frac{I_H}{H} & \alpha_2 S_M \frac{I_H}{H} + \mu I_M \end{array} \right) \Delta t \quad (80)$$

For brevity, the matrix \mathbf{G} is given as,

$$\mathbf{G} = \left(\begin{array}{ccccccc} -\sqrt{\alpha_1 \frac{S_H}{H} I_M} & \sqrt{\theta R_H} & 0 & 0 & 0 & 0 & 0 \\ \sqrt{\alpha_1 \frac{S_H}{H} I_M} & 0 & -\sqrt{\gamma I_H} & 0 & 0 & 0 & 0 \\ 0 & -\sqrt{\theta R_H} & \sqrt{\gamma I_H} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\sqrt{\alpha_2 S_M \frac{I_H}{H}} \sqrt{\mu M} - \sqrt{\mu S_M} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{\alpha_2 S_M \frac{I_H}{H}} & 0 & 0 & -\sqrt{\mu I_M} \end{array} \right). \quad (81)$$

It can be quickly checked to see that this satisfies $\mathbf{G}\mathbf{G}^T = \mathbf{C}$. Utilising this information and plugging back into Equation (77) gives the set of Itô SDEs formulated

in equations (82) to (86).

$$dS_H(t) = \left[-\alpha_1 \frac{S_H}{H} I_M + \theta R_H \right] - \sqrt{\alpha_1 \frac{S_H}{H}} I_M dW_1(t) + \sqrt{\theta R_H} dW_2(t), \quad (82)$$

$$dI_H(t) = \left[\alpha_1 \frac{S_H}{H} I_M - \gamma I_H \right] + \sqrt{\alpha_1 \frac{S_H}{H}} I_M dW_1(t) - \sqrt{\gamma I_H} dW_3(t), \quad (83)$$

$$dR_H(t) = [\gamma I_H - \theta R_H] - \sqrt{\theta R_H} dW_2(t) + \sqrt{\gamma I_H} dW_3(t), \quad (84)$$

$$\begin{aligned} dS_M(t) = & \left[-\alpha_2 S_M \frac{I_H}{H} + \mu M - \mu S_M \right] - \sqrt{\alpha_2 S_M \frac{I_H}{H}} dW_4(t) \\ & + \sqrt{\mu M} dW_5(t) - \sqrt{\mu S_M} dW_6(t), \end{aligned} \quad (85)$$

$$dI_M(t) = \left[\alpha_2 S_M \frac{I_H}{H} - \mu I_M \right] + \sqrt{\alpha_2 S_M \frac{I_H}{H}} dW_4(t) - \sqrt{\mu I_M} dW_7(t). \quad (86)$$

The Wiener processes $dW_i(t) \sim \text{Normal}(0, dt)$. Note, scaled additive noise was used in this scenario, in the form of matrix \mathbf{G} (Allen, 2017), to represent the function $\sigma(X(t))$. Alternatives for the stochastic perturbations is to treat these as constants (plain additive noise), as implemented by Witbooi *et al.* (2020).

Simulations of this model, using Euler-Maruyama numerical approximation can be found in Figure 8 and Figure 9.

4.3 Parameter choices

A crucial part in simulating and evaluating the models is the parameter values. In the baseline models seen later (Sec. 5.1), the parameters are artificially chosen and are not based on real world data or intuitions. To create accurate models, these parameters need to be more carefully selected to represent the shifting dynamics in a region. Targeted approaches and integration of certain control measures may require sourcing of parameters from local data sets, or models based in similar regions. No specific targeted region is focused on in this model, however regions of interest for the spread of malaria are sub-Saharan Africa, South America and South-East Asia. When it comes to spatial, weather dependent, or other models which vary greatly on the region of focus, a more targeted approach is required.

The initial values of parameters can be found in Table 1, which is then subsequently evaluated in numerical simulations in Section 5.2.1. Further to these parameter values, another consideration is the initial populations used in the simulations, as the discrepancy in size can have an effect on the basic reproduction number; in the initial model it is assumed that the mosquito population is around eight times the human population.

Table 1: List of parameter values used in the stochastic malaria model (excluding model adaptations).

Name	Definition	Initial Value	Reference
α_1	Infection rate from mosquito to human, defined as $\alpha_1 = kp$ with probability of infection from a single bite (p) and rate of bites (k)	0.029	Zhang and Lui (2020); Witbooi <i>et al.</i> (2020)
α_2	Infection rate from mosquito to human, defined as $\alpha_2 = kq$ with probability of acquiring malaria from a single bite (q) and rate of bites (k)	0.0156	Zhang and Lui (2020); Witbooi <i>et al.</i> (2020)
γ	Recovery rate of infected individual	0.074	Adjusted from Fatmawati <i>et al.</i> (2020)
μ	Natural birth and death rate of mosquito population	0.033	Averaged from ranges by Zhang and Lui (2020); Chitnis <i>et al.</i> (2008)
θ	Rate at which immunity is lost	1.37×10^{-3}	Averaged from Agusto <i>et al.</i> (2012)

Using the parameters given by Table 1, yields the following \mathcal{R}_0 value in Equation (87).

$$\mathcal{R}_0 = 8 \cdot \left(\frac{(0.029)(0.0156)}{(0.074)(0.033)} \right) \approx 1.5. \quad (87)$$

4.3.1 Sensitivity Analysis: Overview

To identify the parameters whose uncertainty has the largest effect on the model, sensitivity analysis is employed. Outlined by Chitnis *et al.* (2008) and Zamir *et al.* (2016), the key concept in quantifying the uncertainty is the sensitivity index, described by the following Definition 4.1.

Definition 4.1 (Normalised Forward Sensitivity Index). The normalised forward sensitivity index of a variable v , depending on a parameter p is given by Equation (88):

$$\Upsilon_p^v = \frac{\partial v}{\partial p} \times \frac{p}{v}. \quad (88)$$

The sensitivity indices of the basic reproduction number \mathcal{R}_0 is the natural choice to evaluate the overall model. Indices are calculated for each parameter which is then used to analyse the parameters that are most likely to have the biggest impact on infection numbers. Note, the rate of infection from mosquito to human and vice versa is expanded into $\alpha_1 = kp$ and $\alpha_2 = kq$.

As an example, consider the sensitivity index of \mathcal{R}_0 with respect to the rate of recovery γ :

$$\begin{aligned}
 \Upsilon_{\gamma}^{\mathcal{R}_0} &= \frac{\partial \mathcal{R}_0}{\partial \gamma} \cdot \frac{\gamma}{\mathcal{R}_0} \\
 &= -\frac{1}{\gamma^2} \left(\frac{M}{H} \right) \left(\frac{k^2 pq}{\mu} \right) \cdot \frac{\gamma}{\mathcal{R}_0} \\
 &= -\frac{\gamma}{\gamma^2} \mathcal{R}_0 \cdot \frac{\gamma}{\mathcal{R}_0} \\
 &= -1.
 \end{aligned} \tag{89}$$

Given the process at which this parameter was derived, it should be clear that other parameters will attain similar sensitivity indexes. For a formal list of the remaining parameters in the initial \mathcal{R}_0 definition given by Equation (74), see Table 2.

Table 2: Sensitivity indices on \mathcal{R}_0 .

Parameter	Sensitivity Index
p	1
q	1
k	2
γ	-1
μ	-1

Interpretation of these results can be conceived as followed: A negative score pertains the event in which as the parameter value is increased the \mathcal{R}_0 will decrease, with a positive score denoting the increase leads instead to the increase of \mathcal{R}_0 . In the example of the recovery rate, which yielded a value of -1, increasing the parameter by say 10%, would then drive down the basic reproduction number by 10%. It can be seen from Table 2, the bite rate is the most sensitive parameter, with all the rest being of equal sensitivity, albeit with two different types of impact. Variations in the model when these parameters are adjusted are found in Section 6. Practical applications a sensitivity index can achieve, is that the results allow researches to formulate mitigation methods to target parameters which impact infections the most (Zamir *et al.*, 2016).

4.4 Model adaptations

Here, different model adaptations are considered. Although the model formulated thus far emulates the direction taken by other, similar models, there are still modifications that can be made and pondered upon. These changes depend on scenarios and events that may not be essential to the general applicability of a model, but instead the tweaks can highlight alternative behaviour an infectious disease model can take. Two adaptations are explored with the first varying the mosquito population size and fitting towards a growth model. The second is the implementation of a control technique to lower the vector population and in turn lower case numbers and the pool of potentially infectious mosquitoes.

4.4.1 Variable mosquito population size

So far, the population of mosquitoes and humans have been assumed constant i.e.

$$\begin{aligned}\frac{dS_H}{dt} + \frac{dI_H}{dt} + \frac{dR_H}{dt} &= 0, \\ \frac{dS_M}{dt} + \frac{dI_M}{dt} &= 0.\end{aligned}$$

However, in systems where population turnover is rapid, or over an extended period of time, this assumption becomes less viable. A solution to this would be expressing the population in terms of a growth model, with a pool of options to choose from. If the currently derived malaria model still assumes a short time period to simulate, the human population is likely not too worth changing without even mentioning that births and deaths are already neglected. Instead, given the short life-cycle of a mosquito population, a change to the initial model to acknowledge a potential fluctuating population is perhaps the most sensible avenue to take. For a well mixed mosquito population, a suitable choice would be to use the logistic growth model (also known by *Verhulst's equation* in literature). The model is described in Equation (90) and simulates a population where competition for resources is prevalent. In turn, this prevents constant exponential growth in the system.

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K}\right), \quad (90)$$

for a growth rate, r , carrying capacity K (not to be confused with the bite rate, k) and population N .

Deterministic formulation

The rate of change in the population is expressed in terms of a birth and death term. Thus, in the most generalised form, the mosquito population is described as follows in Equation (91),

$$\frac{dM}{dt} = b(M) - d(M), \quad (91)$$

in which $b(M)$ and $d(M)$ are birth and death functions respectively. To express this in terms of a logistic growth model, it can be easily derived by expanding the bracket in Equation (90). Noting that the growth rate is expressed similarly in earlier model as μ , population M and carrying capacity K . The results of this are seen in the following Equation (92),

$$\frac{dM}{dt} = \mu M \left(1 - \frac{M}{K}\right) = \mu M - \mu \frac{M^2}{K} = b(M) - d(M). \quad (92)$$

Considering the introduction of this to the Malaria model, to start, all assumptions from the original Malaria model described in Section 4.1 stay consistent, with the exception of the constant mosquito population. Instead, it is described by Equation (92), with,

$$\frac{dS_M}{dt} + \frac{dI_M}{dt} = \frac{dM}{dt}. \quad (93)$$

Clearly this addition requires adaptations to some of the equations defined by (56) to (60), in particular the two ODEs describing the mosquito population (Eq. (59) and (60)). These are now redefined and given by Equations (94) and (95).

$$\frac{dS_M}{dt} = -\alpha_2 S_M \frac{I_H}{H} + b(M) - \frac{S_M}{M} d(M), \quad (94)$$

$$\frac{dI_M}{dt} = \alpha_2 S_M \frac{I_H}{H} - \frac{I_M}{M} d(M). \quad (95)$$

The birth function can be thought of as an immigration rate, with the death function as a proportion of a respective population dying out, hence the coefficients of S_M/M and I_M/M respectively. Given $S_M + I_M = M$ still holds, adding Equations (94) and (95), yields Equation (93) as required.

Consequentially, the change in the model also leads to a change in the basic reproduction number. A relatively simple alteration, it is now given by replacing μ with $d(M)/M$ in Equation (74), resulting in following Equation (96),

$$\begin{aligned} \mathcal{R}_0 &= \left(\frac{M}{H}\right) \left(\frac{\alpha_1 \alpha_2}{\gamma \frac{dM}{M}}\right) \\ &= \left(\frac{M^2}{H}\right) \left(\frac{\alpha_1 \alpha_2}{\gamma \mu \frac{M^2}{K}}\right) \\ &= \left(\frac{1}{H}\right) \left(\frac{\alpha_1 \alpha_2 K}{\gamma \mu}\right). \end{aligned} \quad (96)$$

The disease dies out if $\mathcal{R}_0 < 1$, and a stable endemic equilibrium exists if $\mathcal{R}_0 > 1$. If the carrying capacity is chosen to be the initial Mosquito population, then Equation (96) is identical to the basic reproduction rate in Equation (74).

SDE model

Implementation of this into a stochastic model is relatively straightforward, the long process of derivation enacted in Section 4.1 is shortened here and the final

matrix \mathbf{G} , given initially by Equation (81) is updated to:

$$\mathbf{G} = \begin{pmatrix} -\sqrt{\alpha_1 \frac{S_H}{H} I_M} \sqrt{\theta R_H} & 0 & 0 & 0 & 0 & 0 \\ \sqrt{\alpha_1 \frac{S_H}{H} I_M} & 0 & -\sqrt{\gamma I_H} & 0 & 0 & 0 \\ 0 & -\sqrt{\theta R_H} \sqrt{\gamma I_H} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{\alpha_2 S_M \frac{I_H}{H}} \sqrt{b(M)} - \sqrt{\frac{S_M}{M} d(M)} & 0 & 0 & 0 \\ 0 & 0 & \sqrt{\alpha_2 S_M \frac{I_H}{H}} & 0 & 0 & -\sqrt{\frac{I_M}{M} d(M)} \end{pmatrix}. \quad (97)$$

The system of SDEs describing the malaria model is now given by the Equation set (98), with $b(M)$ and $d(M)$ expanded to their actual values.

$$\begin{aligned} dS_H(t) &= \left[-\alpha_1 \frac{S_H}{H} I_M + \theta R_H \right] - \sqrt{\alpha_1 \frac{S_H}{H} I_M} dW_1(t) + \sqrt{\theta R_H} dW_2(t), \\ dI_H(t) &= \left[\alpha_1 \frac{S_H}{H} I_M - \gamma I_H \right] + \sqrt{\alpha_1 \frac{S_H}{H} I_M} dW_1(t) - \sqrt{\gamma I_H} dW_3(t), \\ dR_H(t) &= [\gamma I_H - \theta R_H] - \sqrt{\theta R_H} dW_2(t) + \sqrt{\gamma I_H} dW_3(t), \\ dS_M(t) &= \left[-\alpha_2 S_M \frac{I_H}{H} + \mu M - \frac{S_M}{M} \mu \frac{M^2}{K} \right] - \sqrt{\alpha_2 S_M \frac{I_H}{H}} dW_4(t) \\ &\quad + \sqrt{\mu M} dW_5(t) - \sqrt{\frac{S_M}{M} \mu \frac{M^2}{K}} dW_6(t), \\ dI_M(t) &= \left[\alpha_2 S_M \frac{I_H}{H} - \frac{I_M}{M} \mu \frac{M^2}{K} \right] + \sqrt{\alpha_2 S_M \frac{I_H}{H}} dW_4(t) - \sqrt{\frac{I_M}{M} \mu \frac{M^2}{K}} dW_7(t). \end{aligned} \quad (98)$$

A similar method could be implemented for the human population, with a different, more complex model to the logistic growth likely warranted, as to capture the population change. The effect a fluctuating mosquito population has on the overall model is simulated in Section 5.2.1, more specifically Figures 11 and 12.

4.4.2 Introduction of Wolbachia to the population

Wolbachia is a bacterium that inhabits insects cells and one of few available methods in the prevention of diseases spread by mosquitoes, primarily the *Aedes aegypti*. As it competes with other viruses in the mosquitoes system, it suppresses more dangerous viruses from replicating in the cell tissues (O'Neill, 2015). A drawback of Wolbachia is that it is not naturally present or infectious of mosquitoes, therefore introduction of lab infected mosquitoes carrying Wolbachia is required. The bacterium can, however, be transmitted vertically from mother to offspring, so female mosquitoes are the ones initially infected. Limitations of this control method are still present as research has shown the potential for Wolbachia infected mosquitoes to die out quickly in environments with high pesticide usage (Garcia *et al.*, 2020).

Introduced historically to cull mosquito populations in the effort to control the spread of Zika and Dengue viruses, research has more recently turned to focus on malaria prevention, with a current study by [Walker *et al.* \(2021\)](#) indicating the viability of the bacterium for malaria prevention and encouraging further research into the matter. A potential model adaptation, exploring the effect of introducing Wolbachia to a population, is now evaluated.

Consider the most generalised form to accommodate a variable population, expressed in Equation (91). The simplest way to introduce depletion of a population from Wolbachia, is to modify $b(M)$ to incorporate a term denoting the reduction of mosquitoes into the system due to the inability to reproduce. A variable population is still assumed, with the natural birth and death rate now split up into μ_b and μ_d respectively. Depletion as a result of Wolbachia is expressed with ω . This leads to the following adapted definition of the population change dM/dt , in Equation (99):

$$\frac{dM}{dt} = (\mu_b - \omega)M - \mu_d M. \quad (99)$$

Incorporating this into the deterministic model, the mosquito ODEs are now expressed in Equations (100) and (101),

$$\frac{dS_M}{dt} = -\alpha_2 S_M \frac{I_H}{H} + (\mu_b - \omega)M - S_M \mu_d, \quad (100)$$

$$\frac{dI_M}{dt} = \alpha_2 S_M \frac{I_H}{H} - I_M \mu_d, \quad (101)$$

with the basic reproduction rate expressed in the following Equation (102):

$$\mathcal{R}_0 = \left(\frac{M}{H} \right) \left(\frac{\alpha_1 \alpha_2}{\gamma \mu_d} \right). \quad (102)$$

For the stochastic model, the set of SDEs is presented in Equation set (103).

$$\begin{aligned} dS_H(t) &= \left[-\alpha_1 \frac{S_H}{H} I_M + \theta R_H \right] - \sqrt{\alpha_1 \frac{S_H}{H} I_M} dW_1(t) + \sqrt{\theta R_H} dW_2(t), \\ dI_H(t) &= \left[\alpha_1 \frac{S_H}{H} I_M - \gamma I_H \right] + \sqrt{\alpha_1 \frac{S_H}{H} I_M} dW_1(t) - \sqrt{\gamma I_H} dW_3(t), \\ dR_H(t) &= [\gamma I_H - \theta R_H] - \sqrt{\theta R_H} dW_2(t) + \sqrt{\gamma I_H} dW_3(t), \\ dS_M(t) &= \left[-\alpha_2 S_M \frac{I_H}{H} + (\mu_b - \omega)M - S_M \mu_d \right] - \sqrt{\alpha_2 S_M \frac{I_H}{H}} dW_4(t) \\ &\quad + \sqrt{\mu_b M} dW_5(t) - \sqrt{S_M \mu_d} dW_6(t) - \sqrt{\omega M} dW_8(t), \\ dI_M(t) &= \left[\alpha_2 S_M \frac{I_H}{H} - \mu_d \right] + \sqrt{\alpha_2 S_M \frac{I_H}{H}} dW_4(t) - \sqrt{\mu_d} dW_7(t). \end{aligned} \quad (103)$$

Parameter choices for μ_b , μ_d and ω are provided in Table 3.

Table 3: Initial values for new parameters.

Name	Definition	Initial Value	Reference
μ_b	The natural birth rate of mosquitoes.	0.033	Same as μ in Table 1.
μ_d	The natural death rate of mosquitoes.	0.033	Same as μ in Table 1.
ω	depletion of birth rate due to Wolbachia	0.004125	Based on a 12.5% reduction in birth rate, averaged from Zhang and Lui (2020) .

Note, in regards to the sensitivity index of μ_d on \mathcal{R}_0 , the value of μ_d is equivalent to that of μ in the original model, specifically a value of -1 . The ω value is based on the current μ_b rate, as it reflects a proportionate reduction to the natural birth rate. Finally, the birth and death rates are initially set as equal. Since there is no available peer-reviewed study into the reduction Wolbachia can have to reduce malaria at the time of writing, the effects are assumed to be akin to that found through the introduction of dengue-carrying mosquitoes.

5 Results

All models in this section have been produced using the **R** programming language, and code can be found in Appendix B. For reproducible results, stochastic models use the same random number seed when the program is initially run, this leads to certain graph depicting similar looking stochastic waves. However, within each graph, each realisation is completely independent of the other.

This section first explores baseline models, which run simulations of the stochastic methods described in Section 3. Afterwards, a plethora of simulated models for malaria are graphed, using Stochastic Differential Equations, simulated with the Euler-Maruyama numerical approximation. Suitable commentary and comparison with all models considered are provided with all simulations carried over 100 days unless stated otherwise.

5.1 Baseline SIR models

The following results are simulations of the SIR model without vital dynamics, performed as an introductory to the more complex vector-borne stochastic models used for malaria.

5.1.1 Chain Binomial Model

The chain binomial model is one of the simplest discrete time stochastic models, and is a useful baseline for highlighting their unpredictable nature. Figure 5 gives an example of a chain binomial model of 7 iterations with an arbitrary $\mathcal{R}_0 = 1.5$ (representing an epidemic outbreak) and a fixed population. Unlike future models to be explored, it can be seen that the chain binomial model dies out quickly with large populations and discrete time steps lead to chunky graphs where conclusions are hard to be drawn. For this reason, chain binomials are perhaps better suited to small populations and calculating the probabilities of spread under small settings, as opposed to larger populations (Bailey, 1964).

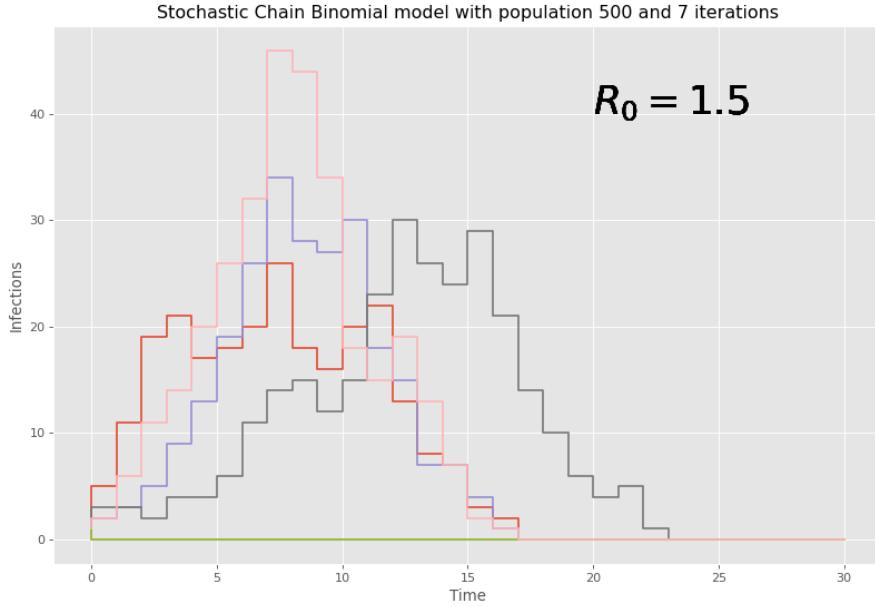


Figure 5: Chain Binomial model recording the number of infections over a 30 day period. Based on the updates and derivations from Equations (22), (23) and (28), the simulation is run 7 times to highlight the stochastic and variable nature of the model with a basic reproduction rate of $\mathcal{R}_0 = 1.5$ to best visualise the progress of an outbreak.

5.1.2 Stochastic simulation with Gillespie's Algorithm

Packages that are used to implement Gillespie's Algorithm acknowledge computational problems with the Direct method and the impracticalities that come with calculating each reaction individually. Instead, a heuristic adaptation is employed, known as tau-leaping (Gillespie, 2001); itself containing several variations. The **R** package *GillespieSSA* (Pineda-Krch, 2009) is one such implementation of Gillespie's Algorithm and despite having the option of Gillespie's Direct method, tau-leaping is recommended for running stochastic simulations. Figure 6 shows a stochastic simulation of the basic SIR model using optimised tau-leaping with the dynamics described in Example 3.2 and $\mathcal{R}_0 = 3$.

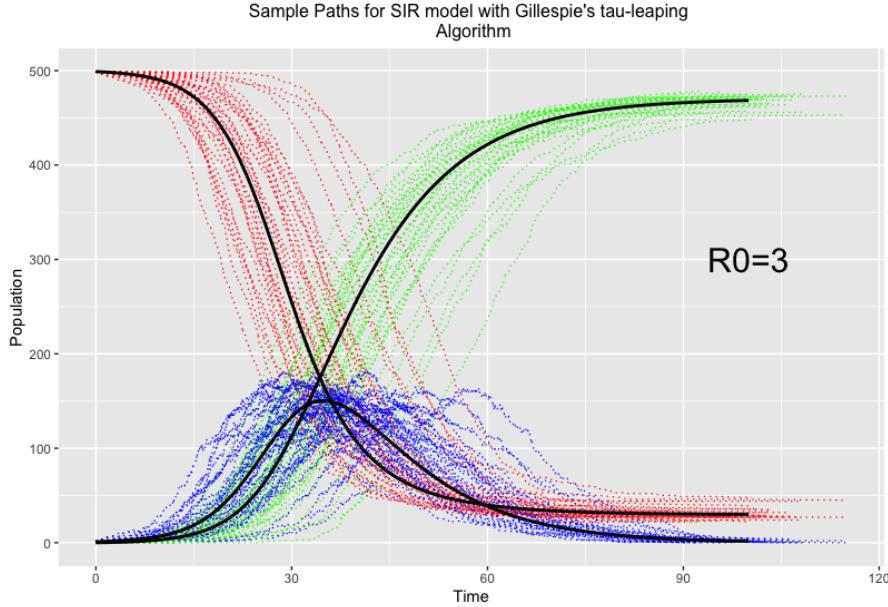


Figure 6: Stochastic simulation of sample paths using Gillespie’s τ -leaping algorithm. An initial population size of 500 with infection and recovery rates of $\beta = 0.3$ and $\gamma = 0.1$ are used, respectively. The coloured dotted lines shows 50 sample paths generated by the algorithm of susceptible (red), infected (blue), and recovered (green) individuals. The thick black lines give the deterministic trajectory for each compartment.

5.1.3 SDE numerical simulation

Simulation using Euler-Maruyama is simple to implement in **R** and can easily be done from scratch without the explicit use of external packages. A slight deviation with the scheme presented in Equation (49) occurs with the generation of the random numbers by Wiener processes. For computational simulations, it is instead best defined by Equation (104),

$$\Delta W = \eta \sqrt{\Delta t}, \quad (104)$$

in which $\eta \sim Normal(0, 1)$ and time discretisation Δt is used.

As seen in Figure 7, each simulation broadly follows the path of the deterministic model. In contrast to Gillespie simulations in Figure 6, there are significantly fewer outliers in the trajectories with the SDE models; alongside more uniformity and smaller variations between each realisation. Albeit the SDE paths still contain enough randomness to accommodate for the real life stochastic nature desired. Due to the optimised time steps taken in Gillespie’s method with tau-leaping, the SDE model does not have these steps, instead they are uniform and set to hourly updates. As a result, a slightly slower computational execution is expected.

Overall, this is likely a suitable numerical method and the requirements to use approximations with higher order terms (Milstein, stochastic Runge-Kutta

etc.) is likely not warranted. As in Gillespie simulation, the parameters are kept the same with $\mathcal{R}_0 = 3$.

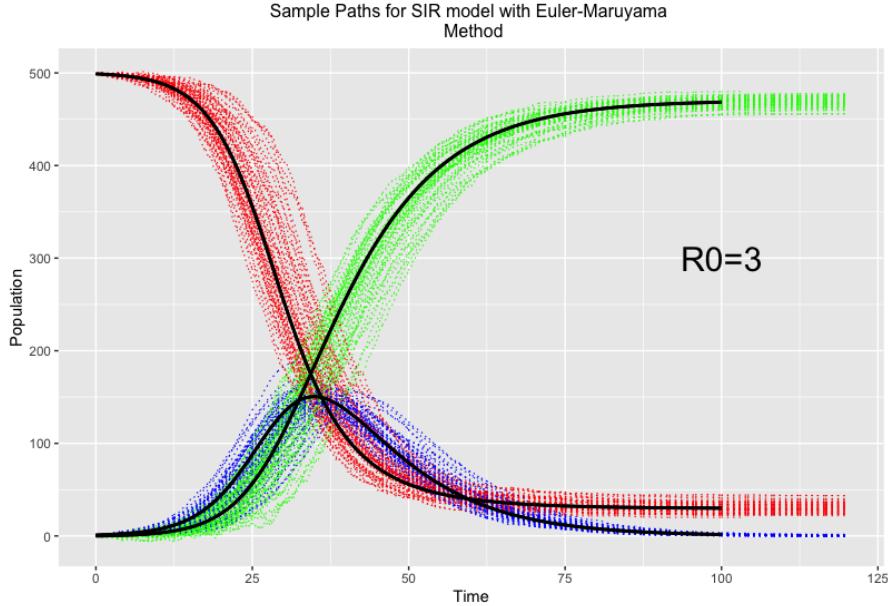


Figure 7: Stochastic simulation of Stochastic Differential Equation of the SIR model without vital dynamics, using the Euler-Maruyama numerical approximation method. An initial population size of 500 with infection and recovery rates of $\beta = 0.3$ and $\gamma = 0.1$ is used, respectively. The dotted lines show individually generated sample paths with susceptible (red), infected (blue), and recovered (green) individuals, and the thick black lines giving the deterministic trajectory of each component.

5.2 Malaria Models

The following malaria models simulate 100 days in a potential epidemic scenario, with all parameters described previously in Tables 1 and 3.

5.2.1 Main malaria model

For the model case study the initial human population is set at $H = 500$ with the mosquito population set at $M = 4000$. There are initially no infected humans and around 100 infected mosquitoes at the start of the epidemic. Figures 8 and 9 give deterministic and 50 stochastic trajectories of the human and mosquito populations, respectively. Concerning the graphing of human compartments (Figure 8), it can be seen that infections maintain a low yet consistent rate, and susceptible and recovered individuals possess rather linear trajectories. Due to the average of a 2 year immunity from malaria, it is no surprise that the recovered population does not decrease or stabilise in the short term. Furthermore, no endemic equilibrium is able to be accurately derived from the first 100 days. With regards to

the mosquito population, the log scale gives insight into its stability of the susceptible mosquito population, and the low infection rate with mosquitoes capable of malaria infection. The simulated paths in both scenarios follow similar to deterministic trajectories, but with enough variation to highlight the effectiveness of these types of models to replicate possible paths an epidemic could take.

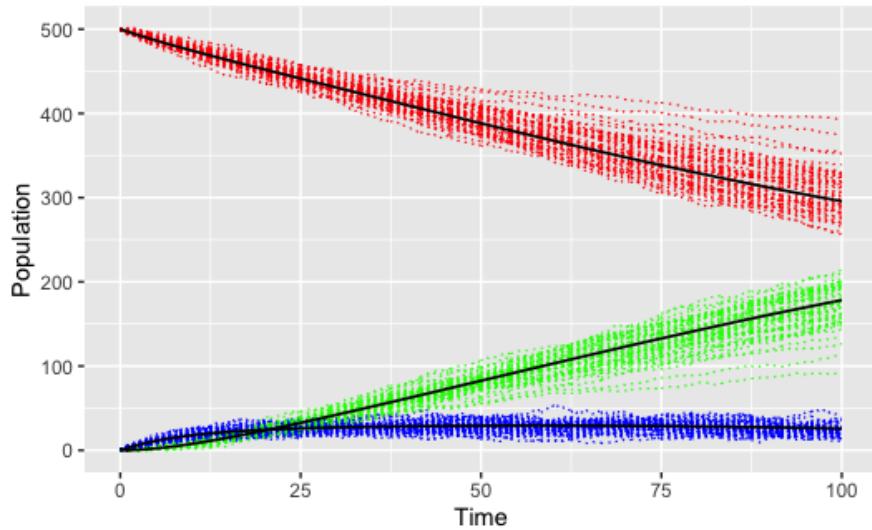


Figure 8: Stochastic simulation of the standard malaria model, showing the change in human compartments over 50 realisations with thick lines denoting deterministic trajectory. Red represents susceptible population, blue infected, and green recovered.

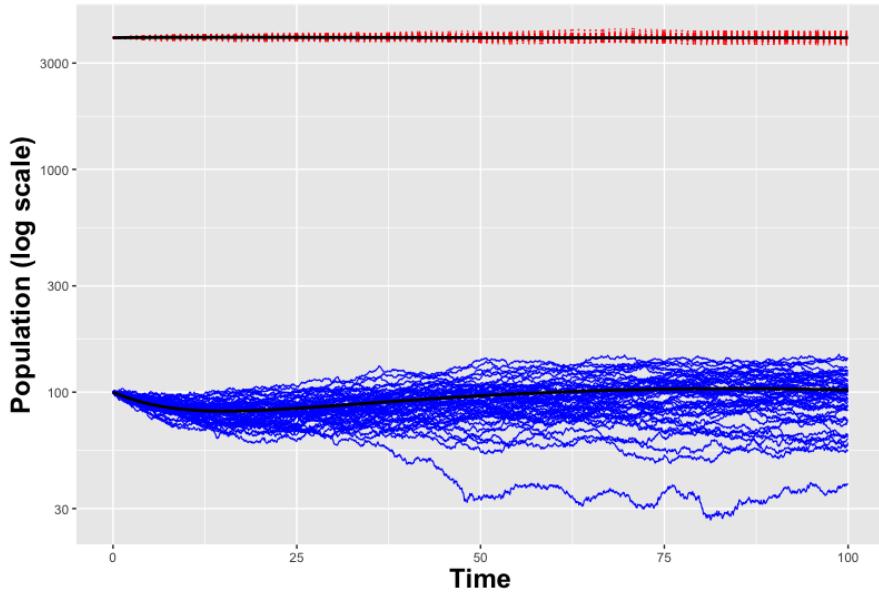


Figure 9: Stochastic simulation of the standard malaria model showing the change in mosquito compartments over 50 realisations, with thick lines denoting deterministic trajectory. The trajectories are plotted on \log_{10} scale with red giving susceptible mosquitoes and blue the infected mosquitoes carrying the *P. falciparum* parasite.

More long term behaviour can also be investigated. Thus far, only 100 days has been considered, but this likely does not cover the entire epidemic process, or end with the final endemic state, therefore a longer time span can be sought. Given human births and deaths are still neglected, it is likely not necessary to simulate many years, as the model would need to reflect shifting human dynamics. Instead, a year is simulated with the resulting human and mosquito compartments graphed in Figure 10. The step size has been reduced to bi-hourly updates, resulting in less uniform trajectories. The potential paths the epidemic can take start varying greatly in the extended time period, however paths in the human compartments begin to stabilise and endemic equilibrium's can be more easily sought after. From Figure 10b, there is a clear downward trend in the infectious mosquito population which can be linked to the endemic state that is starting to be seen in the human population as, with no mosquitoes to infect humans, the disease stifles.

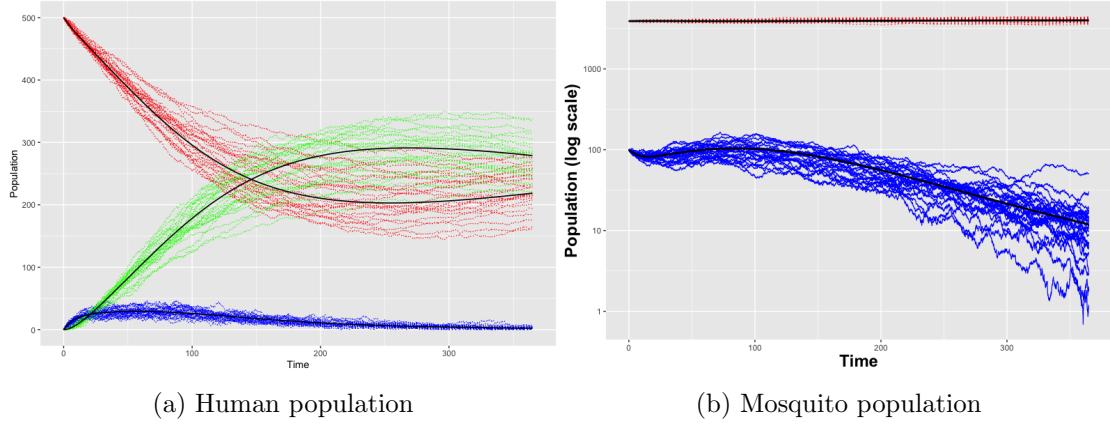


Figure 10: Stochastic simulation of the standard malaria model over a year. For (a) red represents the susceptible population, blue infected, and green recovered. In (b) red represents the susceptible mosquito population and blue the infected mosquitoes carrying the *P. falciparum* parasite.

5.2.2 Malaria models with variable mosquito population

The effect a variable mosquito population can have on the potential trajectory of a malaria epidemic is evaluated in two steps: with an increasing and decreasing populace. As mentioned in Section 4.4.2, the mosquito population follows a logistic growth model with a carrying capacity denoting the maximum colony size. In the graphing of mosquito compartments (Figures 11b and 12b), instead of the susceptible population, the overall population is graphed in its place as well as the carrying capacity. The two models consider the carrying capacity to be reached in all stochastic simulations with the 100 day period a valid choice of simulation length to capture the dynamics effects.

Doubling population capacity

The doubling mosquito population has an initial population of $M = 4000$ and a carrying capacity of $K = 8000$. In Figure 11, it is clear the increasing population increases the number of infectious mosquitoes and in turn the infections of humans, as should be expected. The \mathcal{R}_0 in this model is now 3 and the steepness of the susceptible compartment in Figure 11a is linearly consistent up until 50 days. Afterwards, it exhibits slope akin to the logistic curve. The recovered compartment also clearly changes in steepness with day 25 shifting from a shallow rise to a more linear path. Certain stochastic realisations exert clear distinctions, predominately in the human susceptible and mosquito infectious compartments.

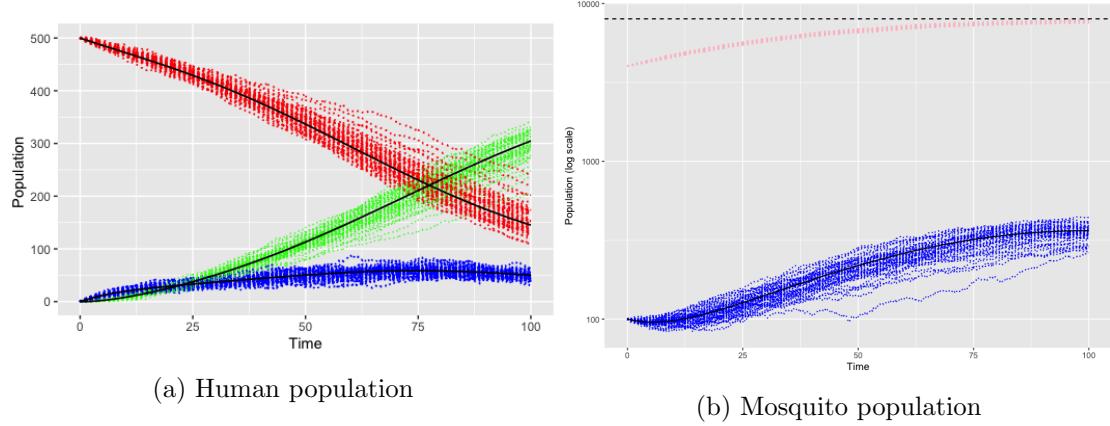


Figure 11: Malaria model with variable mosquito population; initial population starts at $M = 4000$ with a carrying capacity of $K = 8000$. In (a) the red represents a susceptible population, blue an infected and green a recovered human population. In (b) pink gives the total mosquito population, blue the infected mosquito population, and the carrying capacity is represented by the dotted horizontal black line.

Halving population capacity

A halving population starts with the same initial population of $M = 4000$ and a carrying capacity of $K = 2000$. The paths the human compartments take in Figure 12a almost reach an endemic equilibrium status after only 100 days. All stochastic realisations seem consistent and outliers are few and far between. With the mosquito model in Figure 12b, given the log scale, despite clear variations in infectious mosquitoes, these only differ by a few in each realisation. One clear stand out trajectory exists, but again not too much can be read into this.

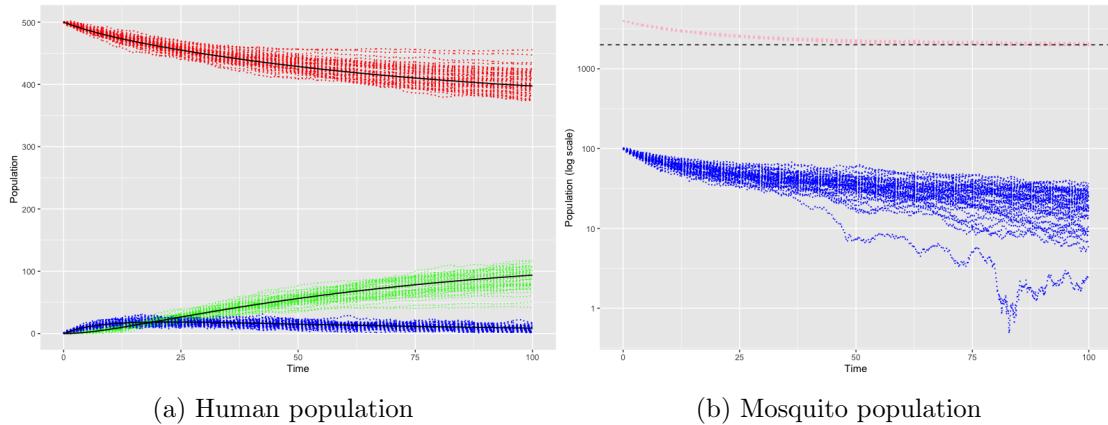


Figure 12: Malaria model with variable mosquito population; initial population starts at $M = 4000$ with a carrying capacity of $K = 2000$. In (a) the red represents a susceptible population, blue an infected and green a recovered population. In (b) pink gives the total mosquito population, blue the infected mosquito population, and the carrying capacity is represented by the dotted horizontal black line.

5.2.3 Malaria model with introduction of Wolbachia

Figure 13 provides the final analysed stochastic model taking into account the effects of reducing the susceptible mosquito population through the introduction of Wolbachia to the female populace. Given the reduction by the bacterium, the model will be expected to have a declining population. In Figure 13b, it can be seen that there is a clear declining susceptible mosquito population, noting that the log scale will make the decline look shallower than it actually is. An initial stabilisation, but subsequent decline, is also present with the infectious mosquitoes. For the susceptible and recovered population in Figure 13a, the stochastic trajectories have a clear distinction, but all clearly show similar trends. The paths are not as stark as seen in other discussed models with declining population (Figure 12) but this assumption is more realistic and based on real data, as opposed to the halving carrying capacity which stays as a theoretical experiment. The infections align with most other models, in which a slight uptick eventually subsides to a stable infection number, albeit with a slight shallow decline.

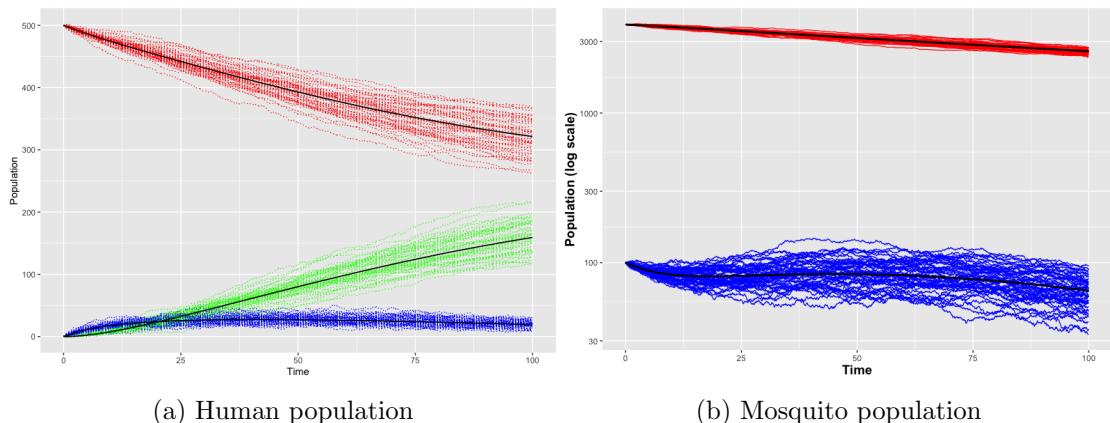


Figure 13: Malaria model with introduction of mosquito population depletion through Wolbachia bacterium. Equal natural birth and death rates are assumed. For (a) red represents the susceptible population, blue infected, and green recovered. In (b) red represents the susceptible mosquito population and blue the infected mosquitoes carrying the *P. falciparum* parasite.

6 Sensitivity Analysis: Results

The following section evaluates the sensitivity of certain model characteristics. The theory behind the analysis is outlined earlier as part of Section 4.3, showcasing the pertinent parameters in the \mathcal{R}_0 equations. As was the case in Section 5, all models are coded in **R** with slight code adaptations, available in Appendix B. Only Stochastic trajectories are considered with multiple realisations for each parameter, this as to avoid misleading results and unknown anomalies.

6.1 Sensitivity of \mathcal{R}_0

A sensitivity analysis on the effects of changing parameters with each index can highlight how the number of infected humans can fluctuate. In the following Table 4, parameters γ (with sensitivity index -1), p (with sensitivity index 1), and k (with sensitivity index 2) are changed with a 50% increase and a 25% decrease from the initial value; the resultant possible \mathcal{R}_0 values within this range are also noted.

Table 4: Changes in parameters for sensitivity analysis. Each parameter is increased by 50% and then decreased by 25% with the resultant ranges the \mathcal{R}_0 value takes between these values noted.

Parameter	Initial value	50% increase	25% decrease	\mathcal{R}_0 range
γ	0.074	0.111	0.0555	$1 - 2$
p	0.07	0.105	0.0525	$1.13 - 2.25$
k	0.42	0.63	0.315	$0.844 - 3.38$

Although these changes are from a theoretical point of view, the reason these parameters could change include the proximity between which humans and mosquitoes interact, leading to a higher (or lower) likelihood of bite rate or disease transmission. The recovery rate, γ , could fluctuate depending on how severe the disease is to a population. For example, if say a human population is dominated by younger or older people, longer recovery times for these vulnerable groups may be apparent.

6.1.1 Main malaria model

Figure 14 graphs the change in the number of infections in the human compartment over a 100 day period. Three realisations for each parameter change is plotted. When varying the parameter k in Figure 14c, a notably significant increase of infections is present at around 50 days when increasing the value (the red lines), which is then followed by a steep decrease and eventual end at a similar point of the red lines in parameter p . This result is consistent with what was seen in the indices of Table 2 where it was shown that k is the most sensitive parameter; a change in its value leads to more dramatic changes compared with other parameters. Figure 14a gives one path when the value is lowered, since this had a negative sensitivity index, a decrease in value indicates an increase in infections. One path follows a potential high number of infections possible, although the other two trajectories demonstrate having more tepid rises, perhaps showing the potential outlier in this stray trajectory and justifying the decision to accommodate multiple stochastic simulations.

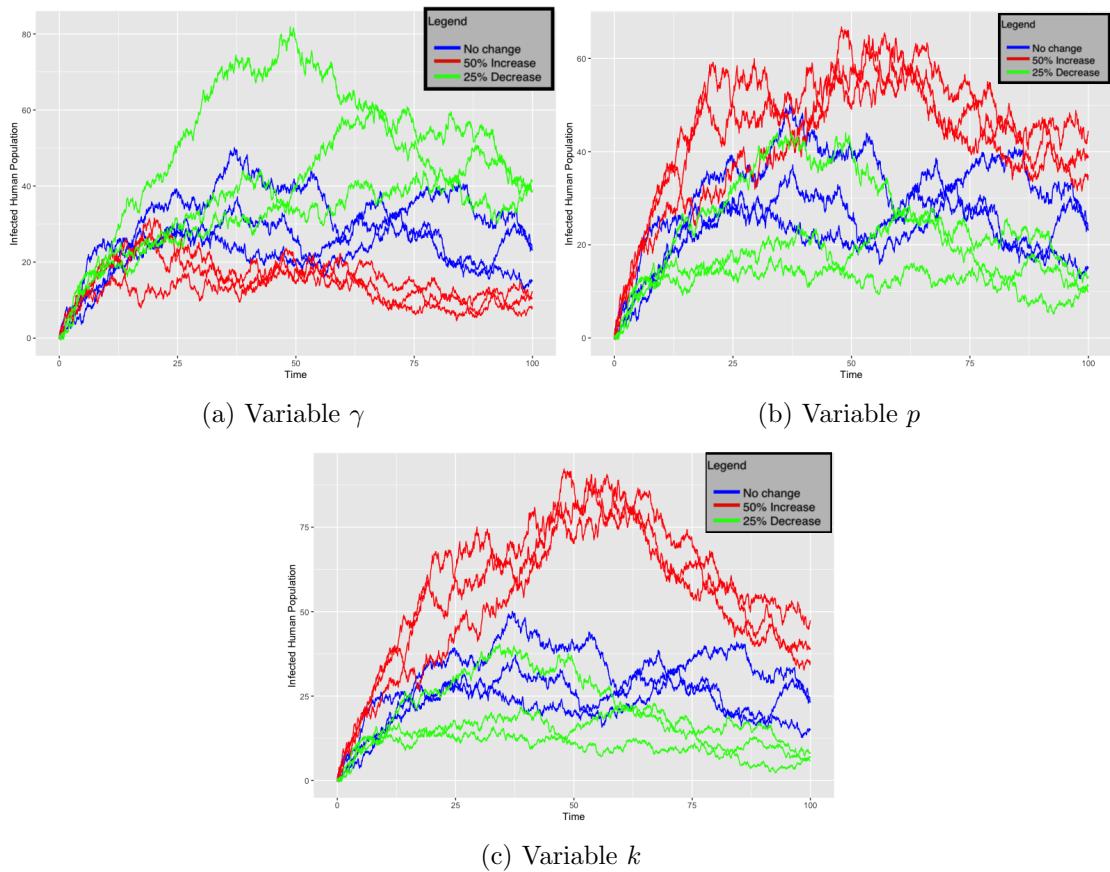


Figure 14: Sensitivity analysis performed on the R_0 parameter by varying the γ , p , and k parameters. The infected human population of the main malaria model is showcased here. Three stochastic realisations of each variation on the parameter is graphed with blue giving the initial value, red the increased value, and green the decreased value.

6.1.2 Adapted models (doubling population)

The sensitivity analysis is also carried out for the two adapted models. Figure 15 shows the event in which the carrying capacity is double the initial mosquito population. Overall, the number of infections, expectedly, reach higher peaks than the main model. There are slight variations between the paths taken in comparison to Figure 14, with similar initial rises. This is followed by equivalently shaped yet higher peaked middle section, before finally giving rise to slight discrepancies in how close and mixed the final number of infected are for each parameter varied. More specifically, it is harder to distinguish, especially when varying p in Figure 15b, any changes in infection numbers by increasing or decreasing the parameter in contrast to the sensitivity of p in the main malaria model where trajectories show different distinguished endpoints.

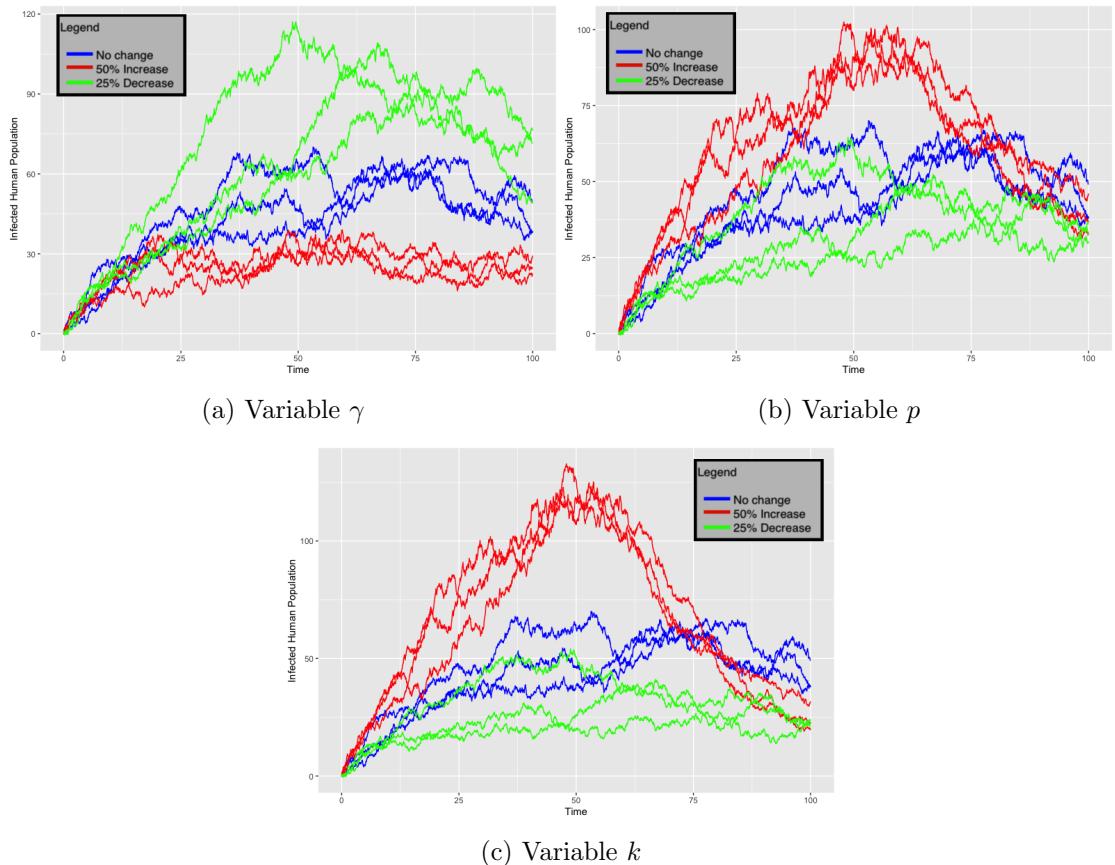


Figure 15: Sensitivity analysis performed on the \mathcal{R}_0 parameter by varying the γ , p , and k parameters and showcasing the infected human population of the adapted malaria model with doubling population capacity ($K = 8000$). Three stochastic realisations of each variation on the parameter is graphed with blue giving the initial value, red the increased value, and green the decreased value.

The adapted model for Wolbachia, evaluating the infected human compartment, is seen in Figure 16 with some significant volatility in results. A point of

interest, not seen much with others, is the dynamic behaviour resulting from varying γ in Figure 16a, especially with the green coloured trajectories. At around 75 days, this decreased value show a significant difference in the number of infections (recalling that due to the sensitivity index of -1 with gamma, the value decreasing leads to the \mathcal{R}_0 value increasing). In the previous two models, the decrease in value of the γ parameter had distinctively higher cases from earlier points, beginning at around 25 days. This is likely due to the delay in a depleting susceptible mosquito population, which lags due to the loss of infectious mosquitoes and then a further delay until this effect is seen in less humans becoming infected. However, eventually when this catches up, the lower recovery rates effect of keeping more people in the infected compartment begins to take effect and eventually distinguishes itself in causing more infected people to arise after 100 days. In the other two models this effect is much quicker to come by. The other two varied parameters (Figures 16b and 16c) highlight the need for multiple stochastic trajectories due to the large case outlier with an increased value in one of the red realisations.

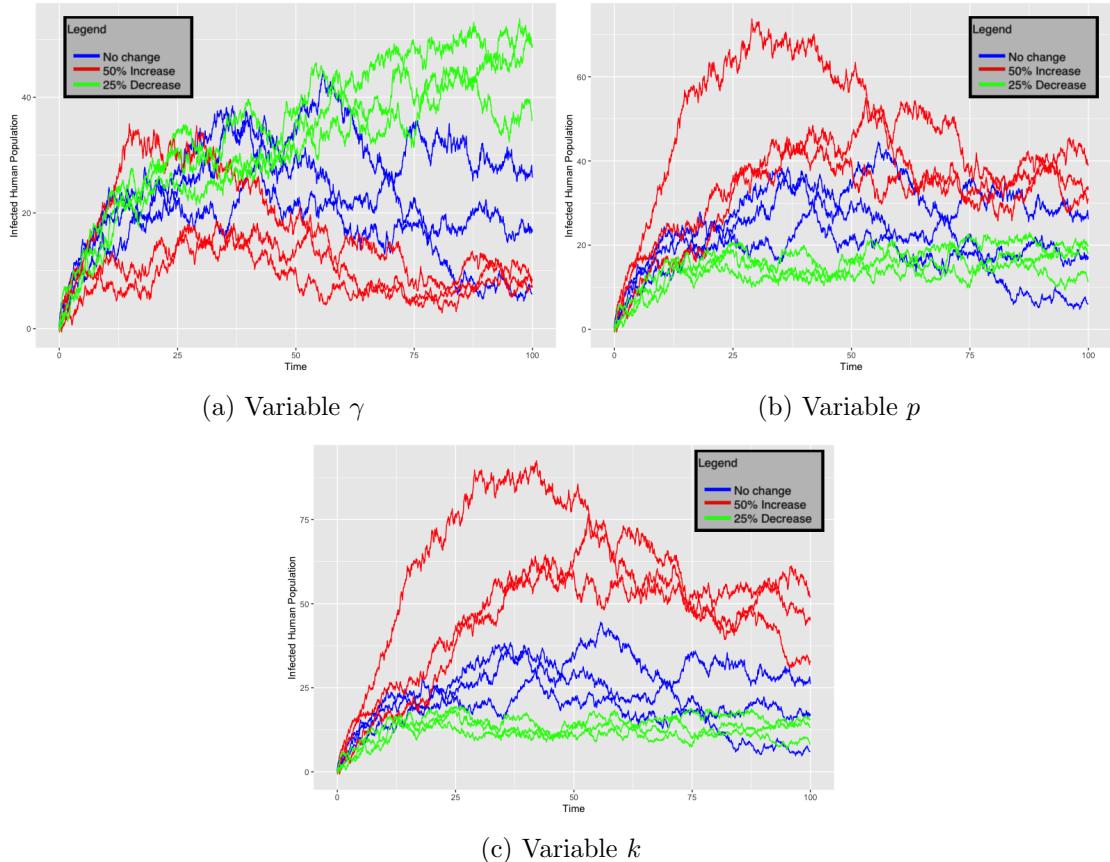


Figure 16: Sensitivity analysis performed on the \mathcal{R}_0 parameter by varying the γ , p , and k parameters and showcasing the infected human population of the adapted malaria model with wolbachia introduced. Three stochastic realisations of each variation on the parameter is graphed with blue giving the initial value, red the increased value, and green the decreased value.

6.2 Sensitivity of other properties

The basic reproduction number is not the only characteristic of the model that can be evaluated through sensitivity analysis: two other interesting areas to explore are the loss of immunity (θ) and depletion through Wolbachia (ω). Both these are parameter values that vary within literature, providing an interesting area to explore further.

6.2.1 Loss of immunity

Immunity from malaria usually lasts around 2 years ([Gonzales et al., 2020](#)). Therefore, the loss of immunity parameter can be calculated at a daily rate of,

$$\frac{1}{365 \cdot 2} \approx 1.37 \times 10^{-3}. \quad (105)$$

This is not the only estimated value of potential loss of immunity. [Chitnis et al. \(2008\)](#) provides a range of potential baseline values using the slow loss of immunity as 5.5×10^{-4} (5 years) and a quick loss as 2.7×10^{-3} (1 year). Implementing these variations in the main malaria model over an annual period as opposed to a 100 day period yields the following for the recovered population as shown in Figure 17. The knock on effects of changing immunity is not apparent until after around 120 days. This leads to the expected result that, over the long run, lower immunity (red line) leads to a smaller recovered/immune population.

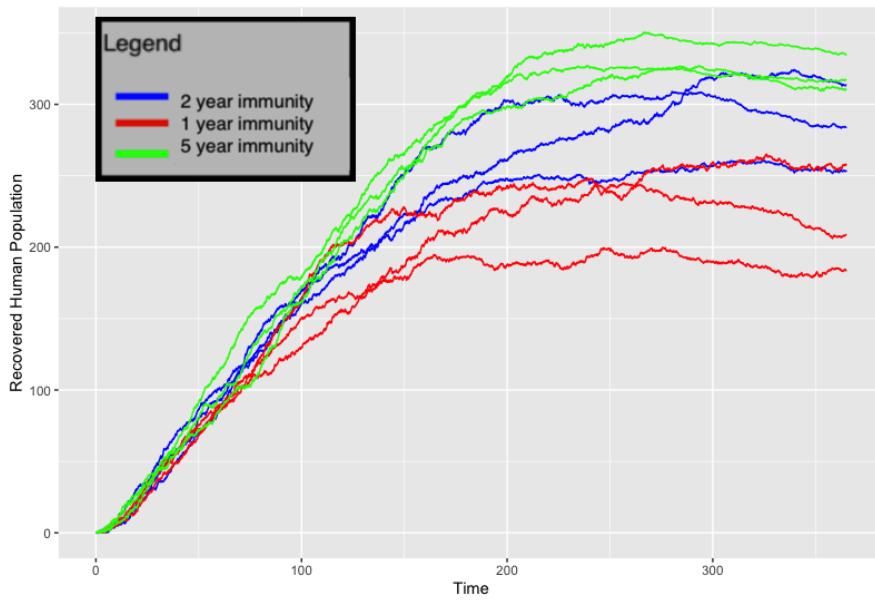


Figure 17: Sensitivity analysis of varying the loss of immunity of the human population for the main malaria model, graphing the number of recovered individuals. Blue trajectory gives realisations of the baseline immunity, green high baseline immunity (5 years) and red the baseline low immunity (1 year).

6.2.2 Wolbachia rate

The susceptible mosquito population does not entirely cover the whole mosquito population but rather covers the mosquitoes that can be infected with malaria and do not carry the Wolbachia bacterium. This assumption is akin to the original model in which only female mosquitoes can become infectious. Using the range given by [Zhang and Lui \(2020\)](#), a baseline low of 0% reduction, and baseline high of 25% reduction is evaluated alongside the average used in the adapted model. The results from this are expressed in Figure 18. Graphing the susceptible mosquito population, the stochastic model sees a quarter reduction in births leading to the population halving in the space of 100 days with one realisation going as far as a final size of 1600 from an initial population of 4000, 3900 of which start off as susceptible. Even with a more modest 12.5% reduction (blue trajectory), a clear and steep reduction is estimated.

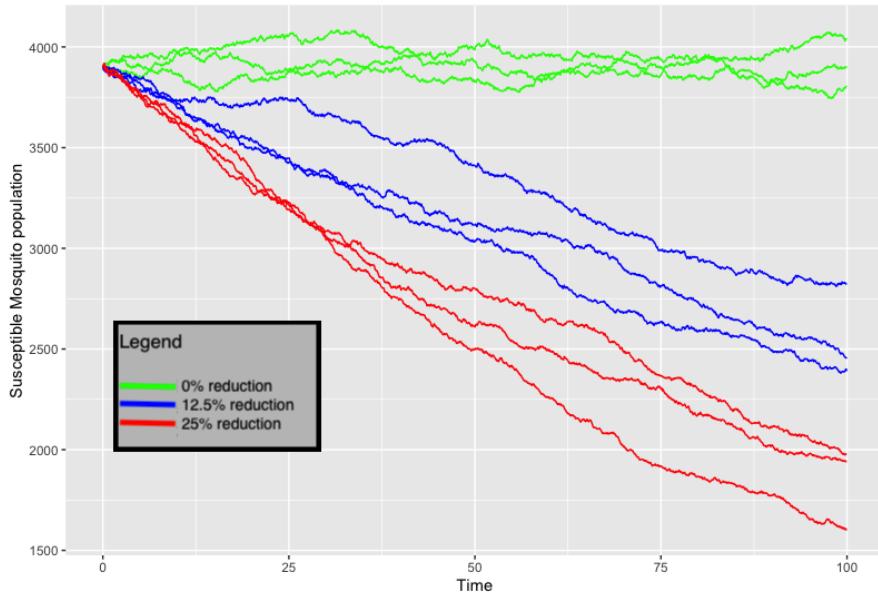


Figure 18: Sensitivity analysis of varying the potential effect Wolbachia has on mosquito population graphing the susceptible mosquito population i.e. those that can be infected by Wolbachia. Blue trajectory gives realisations of the baseline reduction, green no Wolbachia infected mosquitoes, and red with 25% reduction due to Wolbachia infected mosquitoes.

7 Model Validation

Thus far, the data and models shown simulate a theoretical epidemic, and whilst using well sourced parameter values, they currently have no means as to be evaluated for accuracy in real world cases due to a lack of field data. A model evaluated against a simple real world case study allows for pivotal insights. Using the main malaria model, a model validation can be performed against malaria case data from Malaysia.

Bordering to the south of Thailand, the Hulu Perak district is the largest state in Perak, covered in dense rain forest and hills. In 1993, malaria was endemic in the district and accounted for 50% of malaria cases in the entire Perak region ([Rahman et al., 1998](#)). Table 5 gives the monthly number of positive malaria cases in 1993, with the accompanying Figure 19, providing graphical representations for the cumulative number of cases. Using further information from the investigation by [Rahman et al. \(1998\)](#), the population stood at around 90,000 and cases were dominated by the *P. Falciparum* and *P. Vivax* variants of malaria. No mention of mosquito populations were made, but it was acknowledged preventative measures had been introduced to control the population.

Table 5: Total number of positive malaria cases for 1993 in the Hulu Perak district, State of Perak, Malaysia. 332 tested positive in this yearly period.

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Cases	24	25	15	29	24	37	41	47	30	27	18	15

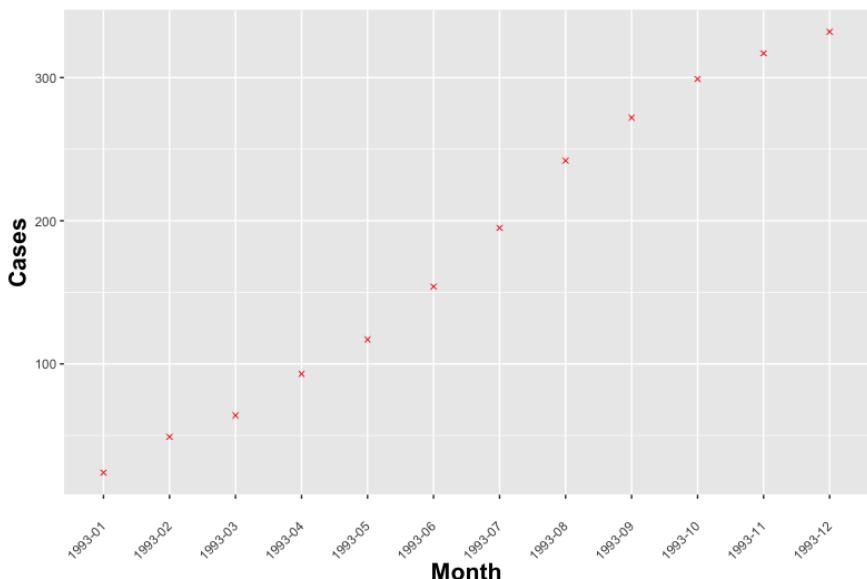


Figure 19: Cumulative number of Malaria cases per month in the Hulu Perak District. The region is said to have been in an endemic malaria state at the time.

For the model validation, all parameter values are consistent with the initial malaria model given in Table 1. The starting human population is $H = 90000$. In the case study, malaria is said to be in an endemic state, therefore $\mathcal{R}_0 \approx 1$. Hence, the \mathcal{R}_0 value is set to 1 for the main malaria model denoted in Equation (74). Rearranging for the mosquito population results in $M = 485820$.

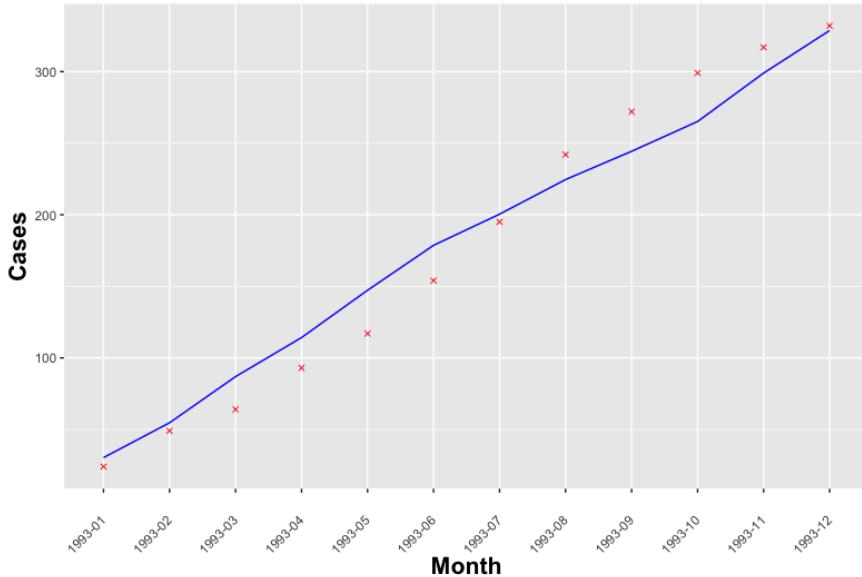


Figure 20: Cumulative number of malaria cases per month in the Hulu Perak District. The red crosses denote case study data and the blue line represents a stochastic trajectory for case numbers using the main malaria model in an endemic state.

The model is evaluated as a single stochastic realisation, using the Euler-Maruyama method. The results are shown in Figure 20. One can also evaluate and compare to the scenario in which the model was not initialised to knowingly have an endemic state. Instead when maintaining the $\mathcal{R}_0 \approx 1.5$ (derived in Section 5), the mosquito population becomes $M = 720000$, and the result of running the same simulation with this initial condition is seen in the supplementary Figure 21. In this case the simulation is an epidemic, but given the case study is in an endemic state, it can be clearly seen that this choice would not be suitable.

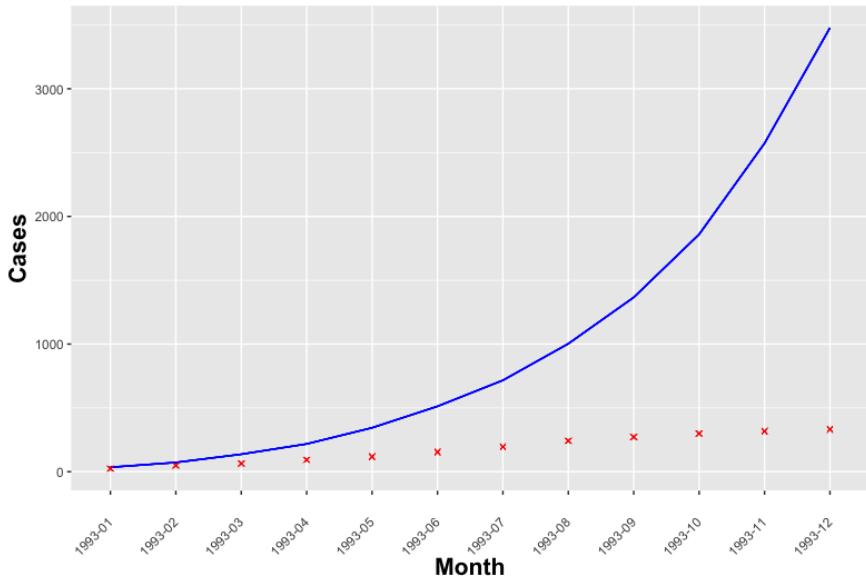


Figure 21: Cumulative number of malaria cases per month in the Hulu Perak District. The red crosses denote case study data and blue line a stochastic trajectory for case numbers using the main malaria model in an epidemic state.

Focusing on Figure 20, it can be seen that this is a pretty suitable fit to the data and likely validates the model as being an apt choice to model a malaria outbreak, given suitable initial condition of an endemic state. When these conditions are not provided i.e. the model follows that of Figure 21, then the model greatly deviates from the case study due to the exponential growth caused by the $\mathcal{R}_0 > 1$.

For a more quantitative analysis, the error between the points and the simulated model for an endemic state can be plotted and analysed. By carrying out 25 stochastic realisations, Figure 22 gives a vertical box plot of the error at each month. Given the cases each month are dependent on that of the previous month, it should be expected that the mean of errors at later months is larger than the earlier months. This is also coupled with the stochastic nature of simulations, leading to some large ranges in errors and a few erroneous outliers. Overall however, the average number of cases at the end of the 12 months appear, on average, to be 100 or so off the actual case study. This gives the model results as roughly 33% higher than expected, while month 7 appears to be the point at which more significant error deviations begin to occur.

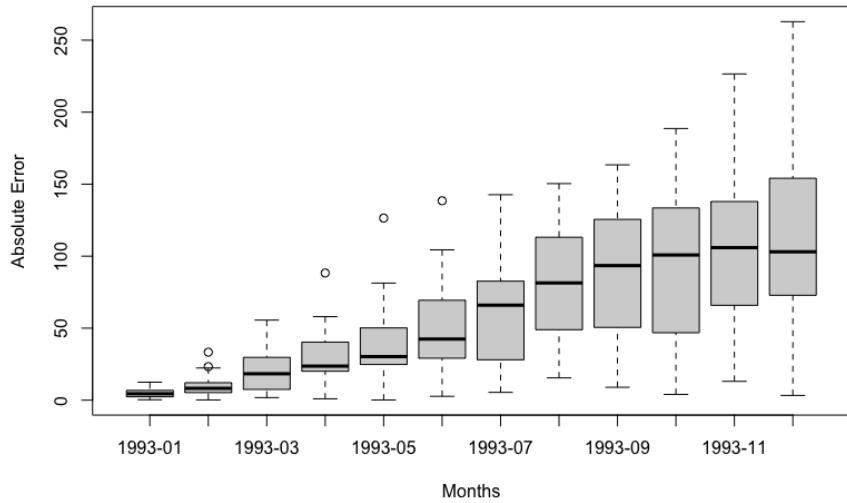


Figure 22: Boxplot of absolute error between case study and 25 stochastic realisations, ordered by month, in an endemic state.

Given this analysis, it is important to not change the variables after comparison to gather a better fit to the case study. Doing so would lead to a potential overfitting of model to the data. Instead, the quantitative and qualitative information should be used to evaluate how this model could be performed better on potentially different and future data sets; implementing changes that affect both execution and methodology. From this model evaluation, it is likely that while the model can give appropriate results, there are still areas that can lead to the minimisation of the error. These include the lack of mitigating parameters built in, lack of consideration for population, and the spatial structure. The parameters perhaps are also in need of better fitting to the area of comparison. More can be said on this, alongside the results found as a whole in the following discussion.

8 Discussion

The main model presented in this thesis aimed to forecast the spread of malaria through a stochastic compartmental system by recording the changes in susceptible, infected, and recovered humans, alongside susceptible and infected mosquitoes. With the plethora of stochastic choices available to build the model, a system of SDEs was eventually chosen and provided useful, consistent results without any overtly outlandish predictions. Some caveats do exist with the model and these are touched upon.

A key advantage of SDEs are that they are based on an underlying deterministic ODE. This makes sure the model broadly follows its deterministic counterpart, and subsequently avoids any major deviations or potential conflicting anomalies. A drawback of this approach is its lack of differentiating features from a deterministic model, exemplified by the stochastic noise being scaled by the square root of the deterministic model. Although the stochastic realisations plotted for the model allowed some variations and gave unique trajectories with each run through, there is likely not a whole lot of discernible information that cannot simply be derived from a deterministic model. Changes that can be made to perhaps give more independence to the SDE model would be to change the coefficient of the Wiener processes. By adding other variations such as constant additive noise or multiplicative noise, this would allow the potential for greater deviations in the results and is something that can be independently investigated. Another avenue to invoke greater changes is evaluation with an alternative numerical approximation scheme such as Milstein or higher stochastic Runga-Kutta methods. Utilising these could produce alternative results, albeit the complexity and necessity of implementation may just lead to this change being a purely theoretical and not a particularly applicable investigation.

Despite the aforementioned dependence between methods, contrasting comparisons can also be made between an ODE and SDE model. Computationally, a stochastic realisation is slower to execute than the deterministic trajectory of the ODE model. On a **2017 model imac with an Intel i5 dual core processor**, one stochastic realisation for the main malaria model, evaluated using the Euler-Maruyama numerical scheme took 2.525 seconds with the overall time for all 50 realisations taking 4 minutes and 35 seconds. With the ODE program, the `desolve` package was used with the default integrator being the `lsoda` method ([Soetaert et al., 2010](#)). An addendum to this is that the ODE solvers' method is likely more optimised than the one used for stochastic realisations. This is due to the stochastic model being evaluated with code created from scratch and following the basic Euler-Maruyama numerical scheme, which contrasts with the ODE solver utilising a regularly maintained software package ported from the (rather) archaic FORTRAN language. Although this accounts for some time discrepancies, another major factor is the larger interval step sizes, and constant random number generation in the Stochastic model. For models requiring longer time evaluations, or many more stochastic trajectories, this may pose more of a problem to be con-

sidered. However, with the models created, the extra time for stochastic program to run does not provide notable frustration or a significance worthy of concern.

Away from computational comparisons, the overall use of both models gives an idea of which scenarios are appropriate for either method. Focusing primarily on an SDE model, the main use case for adding stochastic uncertainty to deterministic trajectories is to map potential scenarios for events in which unexpected deviations or fluctuations can occur. In epidemiological modelling, especially when dealing with large populations that do not act in unison, uncertainty and peculiar results can be expected. With this, a deterministic model, in which the same scenario will play out given the same initial conditions, it is not very likely that the population and evolution of an epidemic will follow this path with precision. One way to combat this would be with confidence intervals and varying the parameters to provide a range of potential results. In contrast, the stochastic model, when executed many times, will always provide different eventualities and give potential situations when a certain degree of randomness is present. From this, many trajectories can be mapped to give their own (in a very liberally described way) confidence intervals on how the infectious disease could play out. While SDE models are perhaps even better suited in areas such as Finance, in which most of the literature is based, epidemiology can also utilise the model with a strong underlying deterministic system to base itself on. This is seen with the model formulation, which as noted earlier can be easily derived and does not require much external additional theory. Other stochastic models such as SSA algorithms (Gillespie) and discrete-time markov chain models (Chain Binomial for example) require complete reformulations and rephrasing of the problem to carry out their simulations. Overall, the SDEs and ODEs can be useful accompaniments to each other. Albeit these alternative stochastic methods can offer insights into the final epidemic size, probability of extinction and epidemic duration; some of which were discussed earlier in the thesis and others can be found in the collection of lecture notes by [Allen \(2008\)](#).

Another evaluation can be made on the model validation. Given that the main model performed well on the model validation, it is reasonable to suggest the adapted models would not add significant improvements. Instead, these would serve the purpose of modelling specific and idiosyncratic events such as the introduction of Wolbachia controls. Moving forward, the use of the model validation then allows out of sample forecasts to be made for the future. Given that the error in the model fitted to the case study is low or can be adjusted for, this then allows reasonable future predictions to be made. As a quick example, Figure 23 plots another year of estimated cases based on the model used in Section 7.

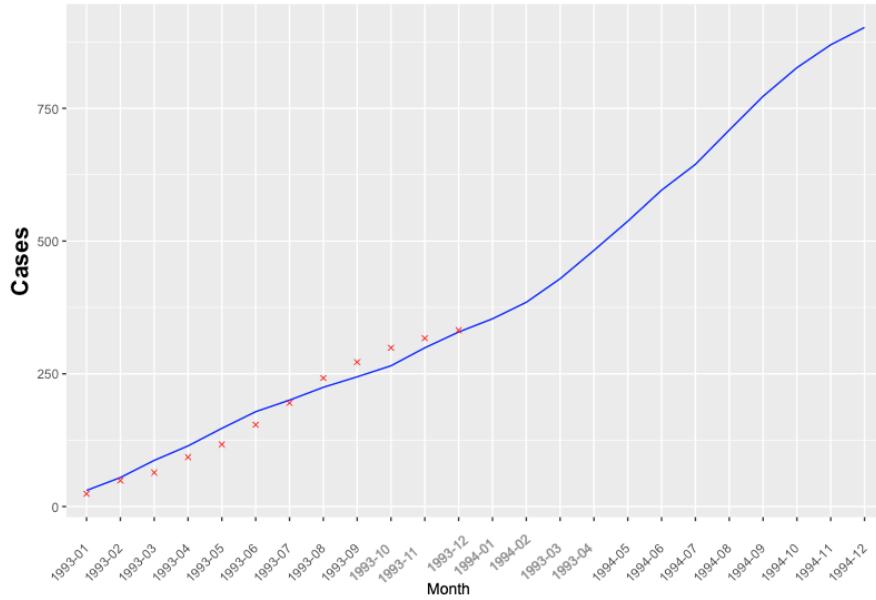


Figure 23: Extended look at cumulative case trajectory for another year of dates.

As deterministic models have historically been preferred, with Agent-based models a current popular direction, discussions are to be had on what these mean for stochastic compartmental models and why these are not currently a popular tool in disease modelling. Stochastic processes and elements will always have some part in disease modelling as uncertainty and randomness will always be a large feature in how diseases travel due to the number of moving parts and assumptions that cannot lead to completely accurate predictions with deterministic models. However, stochastic compartmental stochastic modelling and SDEs may not be a powerhouse methodology in the field moving forward due to a lack of uniqueness and differentiation from other methodologies that outperform on accuracy and speed.

9 Conclusion

The threat malaria poses in many regions is still greatly significant to this day and mathematical models inform on the potential trajectories and the path an outbreak can take. Models also draw attention to parameters and features of the spread of malaria that play a major role, and thus providing governments the basis to accumulate policy decisions that can mitigate and minimise the effects of malaria.

While numerous potential stochastic epidemic models were suggested, the main model in this thesis took the form of a system of SDEs. By using an underlying deterministic system, stochastic qualities were added through scaled additive noise with the primary benefit of using this stochastic technique being the ability to reflect and incorporate uncertainty and randomness into real life systems; something that is not available with deterministic models. However, these stochastic trajectories only offered conservative alternative views, broadly aligning with its deterministic counterpart. Adaptations to the model gave versatility and highlighted an ease of extension to the initial model. A variable mosquito rate removed the strong - and unlikely - assumption the initial model had regarding a constant mosquito population. Through the addition of a logistic growth model, the population could more accurately reflect what is expected and experienced by an insect population. Control measures were realised in the model with the introduction of Wolbachia, a bacterium seen to be useful in the reduction of other vector-borne diseases such as Zika and Dengue viruses.

Concerning other methodologies, alternative approaches to SDE stochastic modelling such as Chain binomial and stochastic simulation algorithms gave an ample baseline view of the potential stochastic models have. The suitability of these likely lag behind that of SDE models. The branching process gave an alternative view for the possible investigations that can be taken with stochastic modelling. A Bienaym  -Galton-Watson process presented a way to predict if an epidemic will die out or persist, with a corresponding probability to this occurring.

For the results, more specifically the SDE model, various numerical approximation methods were explored for the SDE with the eventual preference being the Euler-Maruyama scheme. This was found to have a suitable balance of simplicity, speed and accuracy. The graphs of human and mosquito compartments throughout an epidemic process, accompanied by 50 stochastic realisations and a deterministic path, gave key insights into how malaria can persist in a population. The simulated adapted models showed differences that a doubling mosquito capacity or Wolbachia introduction, among some others, can have in a theoretical epidemic.

The investigation was furthered by performing sensitivity analysis and delving into the role different parameters play on different compartments and how much individual parameters can influence the overall trajectory. In turn, this provides evidence to support policy decisions or targeted approaches that can reduce the severity of the spread of diseases. In the main malaria model, the bite rate was found to be the most pertinent parameter on the number of infections. The rate

of Wolbachia, based on the boundaries suggested by contemporary literature, also gave significant results into how even modest approximations of its effects can have in the collapse of a susceptible mosquito population. A model validation was also performed by analysing how well the model fit a real-life case study of malaria in an endemic region of Malaysia. With some prior information from the region incorporated into the malaria model, the simulation concluded that, the model, whilst prone to erroneous predictions, broadly aligned with the case study and validated the effectiveness and applicability to real life scenarios. From this it can be gathered that mathematical models do have limitations, even with some prior knowledge. These models are best suited to inform and not dictate how an infectious diseases is transmitted through a population.

The area of stochastic compartmental modelling is tiny compared to the overall epidemic field. It is worth remembering that the sub-field only came to be in academic literature in the mid-20th century and with that, rather simple models providing little practical use. This of course has evolved with resources and literature developing stochastic epidemic models and progressing at a quicker rate in recent times. [Allen \(2008, 2017\)](#) provides substantial background and research into stochastic modelling and is a great influence in the understanding that went behind this thesis. For malaria models, the literature has been plentiful and consistent since the initial work by [Ross \(1910\)](#). Further to the work covered so far, the literature also yields ideas into work and improvements that could be made to the models derived, as well as natural extensions.

10 Future directions

Some ideas in which the stochastic model is employed, both applicable to the stochastic nature and modelling in general, are described as follows.

Climate: The first major idea would be the inclusion of variables and parameters introducing weather and climate to the model. Factors such as rainfall have shown to have a general relationship with the mosquito population ([De Kruijf, 1975](#)) so if a malaria model was used for specific regions over time periods where the climates can quickly shift. Tropical climates in which a model is run over overlapping periods between the dry and rainy seasons could benefit from the introduction of climate focused models. A recent model of this ilk, tested on data from central Ethiopia, has been developed by [Gashaw et al. \(2019\)](#).

Spread between geographical locations: Spatial models are also a natural expansion and would introduce a good amount of new theory, alongside expanding on the ideas presented in the thesis. Deterministic spatial models, where populations are expressed in terms of PDEs and subsequently solved numerically, are useful for scenarios where large areas of diverse or sparse populations exist in which the assumption of a well-mixed population cannot be logically applied. To stay on the trend of stochastic modelling, stochastic spatial models (SSMs) are another family within this subset, borrowing theory from statistical mechanics. SSMs can come in the form of a voter model or a grid model. The latter concerns building

an n -dimensional lattice on \mathbb{Z}^n with individual states that then change depending on the surrounding states. The theory and applications of this model type stem from work by [Durrett and Levin \(1997\)](#) and would provide a healthy foundation to expand the current thesis knowledge into stochastic modelling to a spatial structure. In order to keep the compartmental element, meta-population models with each node representing its own compartment can be implemented.

Improved Wolbachia modelling: A final expansion on the model would be to improve how Wolbachia is incorporated into the model. Although the current methodology of adding a single term is not an unreasonable choice, other models take a more complex route. In a paper by [Zhang and Lui \(2020\)](#) working on the spread of Dengue, Wolbachia is modelled as its own independent ODE representing mosquitoes carrying the bacterium. Advantages of this would come with the ability to have better control over, and a greater influence of, the mosquito population. These Wolbachia infected mosquitoes are still present in the system and can interact with others. This is in contrast with the current approach which neglects any potential Wolbachia infected mosquitoes from the system. On the other hand, it leads to complete reformulation of the SDE model, alongside the \mathcal{R}_0 value depending on which compartment infected mosquitoes leave from. Finally, the overall characteristics of the model would need evaluating if these change. Further investigations could be made to compare how well both techniques perform and if there are any discernible discrepancies between the two. Albeit, due to the infancy of research into Wolbachia's ability to control malaria, comparisons to real life case studies are likely not available in the present situation.

References

- Adam, D. (2020), “Special report: The simulations driving the world’s response to covid-19”, *Nature* **580**(7803), 316–318.
- Agusto, F., Marcus, N. and Okosun, K. O. (2012), “Application of optimal control to the epidemiology of malaria”, *Electronic Journal of Differential Equations* **2012**, 1–22.
- Albertsen, K. (1995), “The extinction of families”, *International Statistical Review / Revue Internationale de Statistique* **63**(2).
- Allen, L. (2010), *An Introduction to Stochastic Processes with Applications to Biology*, CRC Press.
- Allen, L. J. S. (2008), *An Introduction to Stochastic Epidemic Models*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 81–130.
- Allen, L. J. S. (2017), “A primer on stochastic epidemic models: Formulation, numerical simulation, and analysis”, *Infect Dis Model* **2**(2), 128–142.
- Alnafisah, Y. (2018), “The implementation of milstein scheme in two-dimensional sdes using the fourier method”, *Abstract and Applied Analysis* **2018**, 1–7.
- Anderson, R. M. and May, R. M. (1981), “The population dynamics of microparasites and their invertebrate hosts”, *Philosophical Transactions of the Royal Society of London. B, Biological Sciences* **291**(1054), 451–524.
- Anderson, R. M. and May, R. M. (1992), *infectious diseases of humans: Dynamics and control*, Oxford University Press, London, England.
- Antia, R., Regoes, R. R., Koella, J. C. and Bergstrom, C. T. (2003), “The role of evolution in the emergence of infectious diseases”, *Nature* **426**(6967), 658–61.
- Aron, J. and May, R. (1982), *The population dynamics of malaria*, Springer, Boston, MA, pp. 139–179.
- Bacaër, N. (2011a), *Daniel Bernoulli, d’Alembert and the inoculation of smallpox (1760)*, Springer London, London, pp. 21–30.
- Bacaër, N. (2011b), *Ross and malaria (1911)*, Springer London, London, pp. 65–69.
- Bailey, N. (1964), *The Elements of Stochastic Processes with Applications to the Natural Sciences*, Wiley.
- Baudoin, F. (2019), *Diffusion Processes and Stochastic Calculus*, EMS textbooks in mathematics, European Mathematical Society, pp. 189–192.
- Becker, N. (1981), “A general chain binomial model for infectious diseases”, *Biometrics* **37**(2).

- Brauer, F., Castillo-Chavez, C. and Feng, Z. (2019), *Introduction: A Prelude to Mathematical Epidemiology*, Vol. 69, Springer, New York, NY, book section 1, pp. 3–19.
- Chitnis, N., Hyman, J. M. and Cushing, J. M. (2008), “Determining important parameters in the spread of malaria through the sensitivity analysis of a mathematical model”, *Bull Math Biol* **70**(5), 1272–96.
- Daston, L. J. (1979), “D’alembert’s critique of probability theory”, *História Mathematica* **6**(3), 259–279.
- De Kruijf, H. A. M. (1975), *The Relation Between Rainfall and Mosquito Populations*, Ecological Studies, Springer, Berlin, Heidelberg, book section Chapter 6, pp. 61–65.
- De Oliveira Costa Maia, J. (1952), “Some mathematical developments on the epidemic theory formulated by reed and frost”, *Human Biology* **24**(3), 167–200.
- DeGroot, M. and Schervish, M. (2012), *Probability and Statistics*, Pearson Education, Incorporated.
- Dickens, B. L., Yang, J., Cook, A. R. and Carrasco, L. R. (2016), “Time to empower release of insects carrying a dominant lethal and wolbachia against zika”, *Open Forum Infect Dis* **3**(2), ofw103.
- Diekmann, O., Heesterbeek, J. A. and Metz, J. A. (1990), “On the definition and the computation of the basic reproduction ratio r_0 in models for infectious diseases in heterogeneous populations”, *J Math Biol* **28**(4), 365–82.
- Dietz, K. and Heesterbeek, J. A. (2002), “Daniel bernoulli’s epidemiological model revisited”, *Math Biosci* **180**, 1–21.
- Doolan, D. L., Dobano, C. and Baird, J. K. (2009), “Acquired immunity to malaria”, *Clin Microbiol Rev* **22**(1), 13–36, Table of Contents.
- Durrett, R. and Levin, S. A. (1997), “Stochastic spatial models: A user’s guide to ecological applications”, *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* **343**(1305), 329–350.
- Fatmawati, Purwati, U. D. and Nainggolan, J. (2020), “Parameter estimation and sensitivity analysis of malaria model”, *Journal of Physics: Conference Series* **1490**.
- Ferguson, N. (2020), “Covidsim”, <https://github.com/mrc-ide/covid-sim>. [Accessed June 14th, 2021].
- Fine, P. E. (1977), “A commentary on the mechanical analogue to the reed-frost epidemic model”, *Am J Epidemiol* **106**(2), 87–100.

- Frauenthal, J. C. (1980), *Chain Binomial Models*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 41–53.
- Frost, W. H. (1928), “Infection, immunity and disease in the epidemiology of diphtheria, with special reference to some studies in baltimore”, *J Prev Med* **2**(4), 325–343.
- Gani, J. and Jerwood, D. (1971), “Markov chain methods in chain binomial epidemic models”, *Biometrics* **27**(3).
- Garcia, G. A., Hoffmann, A. A., Maciel-de Freitas, R. and Villela, D. A. M. (2020), “Aedes aegypti insecticide resistance underlies the success (and failure) of wolbachia population replacement”, *Sci Rep* **10**(1), 63.
- Gashaw, K. W., Kassa, S. M. and Ouifki, R. (2019), “Climate-dependent malaria disease transmission model and its analysis”, *International Journal of Biomathematics* **12**(08).
- Gatton, M. L. and Cheng, Q. (2004), “Modeling the development of acquired clinical immunity to plasmodium falciparum malaria”, *Infect Immun* **72**(11), 6538–45.
- Gillespie, D. T. (1976), “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions”, *Journal of Computational Physics* **22**(4), 403–434.
- Gillespie, D. T. (1977), “Exact stochastic simulation of coupled chemical reactions”, *The Journal of Physical Chemistry* **81**(25), 2340–2361.
- Gillespie, D. T. (2001), “Approximate accelerated stochastic simulation of chemically reacting systems”, *The Journal of Chemical Physics* **115**(4), 1716–1733.
- Gonzales, S. J., Reyes, R. A., Braddom, A. E., Batugedara, G., Bol, S. and Bunnik, E. M. (2020), “Naturally acquired humoral immunity against plasmodium falciparum malaria”, *Front Immunol* **11**, 594653.
- Greenwood, M. (1931), “On the statistical measure of infectiousness”, *J Hyg (Lond)* **31**(3), 336–51.
- Guidoum, A. C. and Boukhetala, K. (2020), “Performing parallel monte carlo and moment equations methods for itô and stratonovich stochastic differential systems: R package Sim.DiffProc”, *Journal of Statistical Software* **96**(2), 1–82.
- Heesterbeek, J. A. and Roberts, M. G. (2015), “How mathematical epidemiology became a field of biology: a commentary on anderson and may (1981) ‘the population dynamics of microparasites and their invertebrate hosts’”, *Philos Trans R Soc Lond B Biol Sci* **370**(1666).
- Iacus, S. M. (2008), *Simulation and Inference for Stochastic Differential Equations*, Springer Series in Statistics, Springer.

- Kendall, D. G. (1966), “Branching processes since 1873”, *Journal of the London Mathematical Society* **s1-41**(1), 385–406.
- Kermack, W. O., McKendrick, A. G. and Walker, G. T. (1927), “A contribution to the mathematical theory of epidemics”, *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **115**(772), 700–721.
- Kermack, W. O., McKendrick, A. G. and Walker, G. T. (1932), “Contributions to the mathematical theory of epidemics. ii.;the problem of endemicity”, *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **138**(834), 55–83.
- Kermack, W. O., McKendrick, A. G. and Walker, G. T. (1933), “Contributions to the mathematical theory of epidemics. iii.;further studies of the problem of endemicity”, *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **141**(843), 94–122.
- Kisling, L. A. and Das, J. M. (2021), *Prevention Strategies*, StatPearls Publishing Copyright 2021, StatPearls Publishing LLC., Treasure Island (FL).
- Le, P. V. V., Kumar, P. and Ruiz, M. O. (2018), “Stochastic lattice-based modelling of malaria dynamics”, *Malaria Journal* **17**(1), 250.
- Lutambi, A. M., Penny, M. A., Smith, T. and Chitnis, N. (2013), “Mathematical modelling of mosquito dispersal in a heterogeneous environment”, *Math Biosci* **241**(2), 198–216.
- Macdonald, G. (1956), “Epidemiological basis of malaria control”, *Bulletin of the World Health Organization* **15**(3-5), 613–626.
- Maki, Y. and Hirose, H. (2013), Infectious disease spread analysis using stochastic differential equations for sir model, in “2013 4th International Conference on Intelligent Systems, Modelling and Simulation”, pp. 152–156.
- Mandal, S., Sarkar, R. R. and Sinha, S. (2011), “Mathematical models of malaria - a review”, *Malaria Journal* **10**(1), 202.
- Matthews, J., Bethel, A. and Osei, G. (2020), “An overview of malarial anopheles mosquito survival estimates in relation to methodology”, *Parasit Vectors* **13**(1), 233.
- Milstein, G. N. (1975), “Approximate integration of stochastic differential equations”, *Theory of Probability & Its Applications* **19**(3), 557–562.
- Molineaux, L., Diebner, H. H., Eichner, M., Collins, W. E., Jeffery, G. M. and Dietz, K. (2001), “Plasmodium falciparum parasitaemia described by a new mathematical model”, *Parasitology* **122**(4), 379–391.
- O’Neill, S. (2015), “The dengue stopper”, *Sci Am* **312**(6), 72–7.

- Pandey, A., Mubayi, A. and Medlock, J. (2013), “Comparing vector–host and sir models for dengue transmission”.
- Pasvol, G. (2006), “The treatment of complicated and severe malaria”, *British Medical Bulletin* **75-76**(1), 29–47.
- Pineda-Krch, M. (2009), “Gillespiessa: A user-friendly stochastic simulation package for r”, *Nature Precedings* .
- Privault, N. (2015), Notes on stochastic finance, in “Notes on Stochastic Finance”, pp. 141–190.
- Rahman, W., Rus, A. and Hassan, A. (1998), “A study on some aspects of the epidemiology of malaria in an endemic district in northern peninsular malaysian near thailand border”, *The Southeast Asian journal of tropical medicine and public health* **29**, 537–40.
- Railsback, S. F. and Grimm, V. (2011), *Agent-Based and Individual-Based Modeling A Practical Introduction*, Princeton University Press.
- Ross, R. (1910), *The prevention of malaria*, Dutton, New York,.
- Ruan, S., Xiao, D. and Beier, J. C. (2008), “On the delayed ross-macdonald model for malaria transmission”, *Bulletin of mathematical biology* **70**(4), 1098–1114.
- Smith, D. L., Battle, K. E., Hay, S. I., Barker, C. M., Scott, T. W. and McKenzie, F. E. (2012), “Ross, macdonald, and a theory for the dynamics and control of mosquito-transmitted pathogens”, *PLoS pathogens* **8**(4), e1002588–e1002588.
- Smith, T., Maire, N., Ross, A., Penny, M., Chitnis, N., Schapira, A., Studer, A., Genton, B., Lengeler, C., Tediosi, F., de Savigny, D. and Tanner, M. (2008), “Towards a comprehensive simulation model of malaria epidemiology and control”, *Parasitology* **135**(13), 1507–16.
- Snow, R. W., Guerra, C. A., Noor, A. M., Myint, H. Y. and Hay, S. I. (2005), “The global distribution of clinical episodes of plasmodium falciparum malaria”, *Nature* **434**(7030), 214–217.
- Sobczyk, K. (1991), *Introduction: Origin of Stochastic Differential Equations*, Springer, book section Chapter 1, pp. 1–5.
- Soetaert, K., Petzoldt, T. and Setzer, R. W. (2010), “Solving differential equations in R: Package deSolve”, *Journal of Statistical Software* **33**(9), 1–25.
- Sridhar, D. and Majumder, M. S. (2020), “Modelling the pandemic”, *BMJ* **369**, m1567.
- Tumwiine, J., Mugisha, J. Y. T. and Luboobi, L. S. (2007), “A mathematical model for the dynamics of malaria in a human host and mosquito vector with temporary immunity”, *Applied Mathematics and Computation* **189**(2), 1953–1965.

- van den Driessche, P. and Watmough, J. (2002), “Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission”, *Mathematical Biosciences* **180**(1-2), 29–48.
- Walker, T., Quek, S., Jeffries, C. L., Bandibabone, J., Dhokiya, V., Bamou, R., Kristan, M., Messenger, L. A., Gidley, A., Hornett, E. A., Anderson, E. R., Cansado-Utrilla, C., Hegde, S., Bantuzeko, C., Stevenson, J. C., Lobo, N. F., Wagstaff, S. C., Nkondjio, C. A., Irish, S. R., Heinz, E. and Hughes, G. L. (2021), “Stable high-density and maternally inherited wolbachia infections in anopheles moucheti and anopheles demeilloni mosquitoes”, *Curr Biol* **31**(11), 2310–2320 e5.
- Wedajo, A. G., Bole, B. K. and Koya, P. R. (2018), “Analysis of sir mathematical model for malaria disease with the inclusion of infected immigrants”, *IOSR Journal of Mathematics* **14**(5), 10–21.
- WHO (2014), “Malaria: fact sheet”, **1**(WHO-EM/MAC/035/E). [Accessed June 14th, 2021].
- Wilkie, J. (2004), “Numerical methods for stochastic differential equations”, *Phys Rev E Stat Nonlin Soft Matter Phys* **70**(1 Pt 2), 017701.
- Wilson, A. L., Courtenay, O., Kelly-Hope, L. A., Scott, T. W., Takken, W., Torr, S. J. and Lindsay, S. W. (2020), “The importance of vector control for the control and elimination of vector-borne diseases”, *PLoS Negl Trop Dis* **14**(1), e0007831.
- Witbooi, P. J., Abiodun, G. J., van Schalkwyk, G. J. and Ahmed, I. H. I. (2020), “Stochastic modeling of a mosquito-borne disease”, *Advances in Difference Equations* **2020**(1).
- Zamir, M., Zaman, G. and Alshomrani, A. S. (2016), “Sensitivity analysis and optimal control of anthroponotic cutaneous leishmania”, *PLoS One* **11**(8), e0160513.
- Zhang, H. and Lui, R. (2020), “Releasing wolbachia-infected aedes aegypti to prevent the spread of dengue virus: A mathematical study”, *Infect Dis Model* **5**, 142–160.

Appendices

A Gillespie Algorithm Variables

Table 6: Variables and parameters used in Gillespie's Direct Algorithm

Name	Definition
\mathbf{X}_0	$X(0)$, a vector storing the initial configuration of the model.
\mathbf{a}	Propensity vector, containing the probabilities that a particular reaction will occur over an infinitesimal time period $[t + \tau, t + \tau + dt]$.
v	State change vector, recording the possible change between different compartments that can occur.
t_{final}	The final time of the model.
r_{final}	The maximum number of reactions (or iterations) allowed.
r_1, r_2	Uniform random numbers generated on the interval $[0, 1]$.
\mathbf{X}	The current state of the configurations for each compartment.
v_j	The column of the state change vector that has been chosen for the algorithm. Used to update \mathbf{X} .
a_0	The sum of all values in the propensity vector, \mathbf{a} .
τ	a randomly determined jump in time, calculated as, $-\frac{1}{a_0} \log(r_1)$.

B Code listings

The following are the R programs used to generate the figures seen in this thesis. Note some have been used to generate multiple figures and the program names should provide suitable information as to what task each program performed.

```

1 # Import SSA algorithm
2 library('GillespieSSA')
3 library(ggplot2)
4
5
6 ## Load deSolve package (For ODE solve)
7 library(deSolve)
8
9 # Set some initial parameters
10 parms <- c(beta=0.3, gamma=0.1)
11 tf <- 100 # Final time
12 x0 <- c(S=499, I=1, R=0)
13 a <- c("(beta/sum(x0))*{S}*{I}", "gamma*{I}")
14 nu <- matrix(c(-1, 0, +1, -1, 0, +1), nrow=3, byrow=T)
15 out<-data.frame(time=seq(1,tf), S=numeric(), I=numeric(), R=numeric())
16 g<-ggplot() # Initialise plotting library
17
18 # Run 50 realisations
19 for (x in seq(1,50)){
20   set.seed(x) # for reproducible results
21   # Run gillespie iteration
22   sim <- ssa(
23     x0 = x0,
24     a = a,
25     nu = nu,
26     parms = parms,
27     tf = tf,
28     method = ssa.otl(),
29     verbose = FALSE,
30     consoleInterval = 1
31   )
32   # Keep updated on iteration
33   print(x)
34
35   # Plot realisation
36   out<-data.frame(time=sim$data[,1], S=sim$data[,2], I=sim$data[,3], R=sim$data[,4])
37   g<-g+geom_line(data=out, aes_string(x="time",y="S"),color='red', lwd = 0.4, linetype='dotted')+geom_line(data=out, aes_string(x="time",y="I"),lwd = 0.4,color='blue',linetype='dotted')+geom_line(data=out, aes_string(x="time",y="R"),lwd = 0.4,color='green',linetype='dotted')
38   print(g)
39 }
40
41 #####
42 #####

```

```

43 # SIR deterministic model - Adapted from: http://rstudio-pubs-
    static.s3.amazonaws.com/6852_c59c5a2e8ea3456abbeb017185de603e.
    html
44 #####
45
46 ## Create an SIR function
47 sir <- function(time, state, parameters) {
48
49   with(as.list(c(state, parameters)), {
50
51     dS <- -(beta * S * I)/(S+I+R)
52     dI <- (beta * S * I)/(S+I+R) - gamma * I
53     dR <- gamma * I
54
55     return(list(c(dS, dI, dR)))
56   })
57 }
58
59 #### Set parameters
60 ## Proportion in each compartment: Susceptible 0.999999, Infected
61 ## 0.000001, Recovered 0
62 init <- c(S = 499, I = 1, R = 0)
63 ## beta: infection parameter; gamma: recovery parameter
64 parameters <- c(beta = 0.3, gamma = 0.1)
65 ## Time frame
66 times <- seq(0, 100, by = 1)
67
68 ## Solve using ode (General Solver for Ordinary Differential
69 ## Equations)
70 out <- ode(y = init, times = times, func = sir, parms = parameters
71 )
72
73 ## change to data frame
74 out <- as.data.frame(out)
75 ## Delete time variable
76 out$time <- NULL
77
78 g<-g+geom_line(data=out, aes_string(x="times",y="S"),color='black',
79 , lwd=1)+geom_line(data=out, aes_string(x="times",y="R"),color=
80 'black', lwd=1)+geom_line(data=out, aes_string(x="times",y="I"),
81 ,color='black', lwd=1)
82 print(g)
83 g<-g+xlab('Time')
84 g<-g+ylab('Population')
85 g+ggtitle("Sample Paths for SIR model with Gillespie's tau-leaping
86 \n Algorithm") +theme(plot.title = element_text(hjust = 0.5,
87 size=12))+annotate("text", x=100, y=300, label= "R0=3",size=8)

```

Listing 1: gillespie.r - Gillespie Algorithm for SIR model

```

1 # Import libraries and set initial conditions
2 library(ggplot2)
3 g<-ggplot()
4 t_init<-0
5 t_final<-100
6 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days

```

```

7 x0 <- c(S=499, I=1, R=0)
8 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
9 params <- c(beta=0.3, gamma=0.1)
10
11 # Euler-Maruyama function
12 em <- function(sir, dt, params) {
13   N <- sum(sir) # Total number in the system
14
15   # Update each compartment (see method in thesis)
16   S_next<- sir['S']+(-(params['beta']*sir['S']*sir['I'])/(N)*dt)+sqrt((params['beta']*sir['S']*sir['I'])/(N))*(rnorm(1)*sqrt(dt))
17   I_next<- sir['I']+((params['beta']*sir['S']*sir['I'])/(N)-params['gamma']*sir['I'])*dt+sqrt((params['beta']*sir['S']*sir['I'])/(N))-sqrt(params['gamma']*sir['I'])*rnorm(1)*sqrt(dt)
18   R_next<- N-S_next-I_next # R is easy to recover given constant population
19
20   return(c(S_next, I_next, R_next)) # Return grouping
21 }
22
23 # Run multiple iterations (50 realisations)
24 for (iter in seq(1,50)) {
25   set.seed(iter) # Set seed for reproducible results
26   print(iter)
27   df<-data.frame(time=numeric(), S=numeric(), I=numeric(), R=numeric()) # Used to store each time point update
28
29   x=x0 # Set initial conditions
30   # Run single iteration
31   for (delta_t in intervals) {
32     updates<-em(x,dt, params) # Run realisation using EM
33
34     # If infections reach 0 break from loop
35     if (updates[2]<= 0) {
36       break
37       print('Exited at time: ', delta_t)
38     }
39
40     df_temp<-data.frame(time=delta_t,S=updates[1],I=updates[2],R=updates[3]) # store in temp. DataFrame
41     df<-rbind(df,df_temp) # Bind to dataframe of all updates
42     x<-updates # Update state variables
43   }
44   # store in output variable then plot
45   out<-df
46   g<-g+geom_line(data=out, aes_string(x="time",y="S"),color='red', lwd = 0.4, linetype='dotted')+geom_line(data=out, aes_string(x="time",y="I"),lwd = 0.4,color='blue',linetype='dotted')+geom_line(data=out, aes_string(x="time",y="R"),lwd = 0.4,color='green',linetype='dotted')
47   print(g) # Needed to show plot
48 }
49

```

```

50 ######
51 # SIR deterministic model
52 #####
53
54 ## Create an SIR function
55 sir <- function(time, state, parameters) {
56
57   with(as.list(c(state, parameters)), {
58
59     dS <- -(beta * S * I)/(S+I+R)
60     dI <- (beta * S * I)/(S+I+R) - gamma * I
61     dR <- gamma * I
62
63     return(list(c(dS, dI, dR)))
64   })
65 }
66
67 ### Set parameters
68 ## Proportion in each compartment: Susceptible 0.999999, Infected
69 ## 0.000001, Recovered 0
70 init <- c(S = 499, I = 1, R = 0)
71 ## beta: infection parameter; gamma: recovery parameter
72 parameters <- c(beta = 0.3, gamma = 0.1)
73 ## Time frame
74 times <- seq(0, 125, by = 1)
75
76 ## Solve using ode (General Solver for Ordinary Differential
77 ## Equations)
78 out <- ode(y = init, times = times, func = sir, parms = parameters
79 )
80
81 ## change to data frame
82 out <- as.data.frame(out)
83 ## Delete time variable
84 out$time <- NULL
85
86 # Plot graphs of deterministic and SDE
87 g<-g+geom_line(data=out, aes_string(x="times",y="S"),color='black',
88 , lwd=1)+geom_line(data=out, aes_string(x="times",y="R"),color=
89 'black', lwd=1)+geom_line(data=out, aes_string(x="times",y="I"),
90 ,color='black', lwd=1)
91 print(g)
92 g<-g+xlab('Time')
93 g<-g+ylab('Population')
94 g+ggtitle("Sample Paths for SIR model with Euler-Maruyama \n
95 Method") + theme(plot.title = element_text(hjust = 0.5, size=12)) +
96 annotate("text", x=100, y=300, label= "R0=3", size=8)

```

Listing 2: SDE.r - Euler-Maruyama simulation for SIR model

```

1 #####
2 # Stochastic malaria model
3 #####
4 library(ggplot2)
5
6 ## Load deSolve package

```

```

7 library(deSolve)
8 set.seed(42)
9 g<-ggplot()
10 h<-ggplot()
11 t_init<-0
12 t_final<-100
13 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days
14 x0 <- c(Sh = 90000, Ih = 0, Rh=0, Sm=450000, Im=11250)
15 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
16 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,
17   theta=1/(365*2))
18
19 # Euler-Maruyama function
20 em <- function(sir, dt, params) {
21   H <- sum(sir[1:3]) # Total human pop
22   M <- sum(sir[4:5]) # total mosquito pop
23   # Predefine wiener processes (this is a crappy way to do it)
24   w1<-rnorm(1)*sqrt(dt)
25   w2<-rnorm(1)*sqrt(dt)
26   w3<-rnorm(1)*sqrt(dt)
27   w4<-rnorm(1)*sqrt(dt)
28   w5<-rnorm(1)*sqrt(dt)
29   w6<-rnorm(1)*sqrt(dt)
30   w7<-rnorm(1)*sqrt(dt)
31
32   # Make sure no negative values
33   if(sir['Ih']<0) {
34     sir['Ih']<-0
35   }
36   if(sir['Rh']<0){
37     sir['Rh']<-0
38   }
39
40   # Update each compartment (see method in thesis)
41
42   # F functions
43   Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params[',
44     'theta']*sir['Rh'])
45   Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params[',
46     'gamma']*sir['Ih'])
47   Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
48   Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+params[',
49     'mu']*M-params['mu']*sir['Sm'])
50   Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-params[',
51     'mu']*sir['Im'])
52
53   # Sigma
54   Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im']
55     ])*w1+sqrt(params['theta']*sir['Rh'])*w2)
56   Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'])
57     *w1-sqrt(params['gamma']*sir['Ih'])*w3)
58   Rh_sigma<-(sqrt(params['gamma']*sir['Ih'])*w3-sqrt(params['theta'
59     ]*sir['Rh'])*w2)
60   Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)
61     ))

```

```

  ) *w4+sqrt(params['mu']*M)*w5-sqrt(params['mu']*sir['Sm'])*w6)
53  Im_sigma<- (sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))
54   *w4-sqrt(params['mu']*sir['Im'])*w7)
# Compute next time step
55 Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
56 Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
57 Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
58 Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
59 Im_next<-sir['Im']+Im_f*dt+Im_sigma
60
61 return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
62 grouping
63 }
64
65 # Run multiple iterations
66 for (iter in seq(1,50)) {
67   set.seed(iter) # Set seed for reproducible results
68   print(iter)
69   df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(), Rh=
70     numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
71   point update
72
73   x=x0 # Set initial conditions
74   # Run single iteration
75   for (delta_t in intervals) {
76     updates<-em(x,dt, params) # Run realisation using EM
77
78     df_temp<-data.frame(time=delta_t, Sh=updates[1], Ih=updates[2],
79     Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in temp.
80     DataFrame
81     df<-rbind(df,df_temp) # Bind to dataframe of all updates
82     x<-updates # Update state variables
83   }
84   # store in output variable then plot
85   out<-df
86
87   g<-g+geom_line(data=out, aes_string(x="time",y="Sh"),color='red',
88     ,lwd = 0.4, linetype='dotted')+geom_line(data=out, aes_string(x
89     ="time",y="Ih"),lwd = 0.4,color='blue',linetype='dotted')+geom_
90     line(data=out, aes_string(x="time",y="Rh"),lwd = 0.4,color='
91     green',linetype='dotted')
92   h<-h+geom_line(data=out, aes_string(x="time",y="Sm"),color='red'
93     ,lwd = 0.4, linetype='dotted')+geom_line(data=out, aes_string(x
94     ="time",y="Im"),lwd = 0.4,color='blue')
95   print(g) # Needed to show plot
96   print(h)
97 }
98
99 #####
100 # Malaria deterministic model
101 #####
102
103 ## Create an SIR function
104 sir <- function(time, state, parameters) {

```

```

94   with(as.list(c(state, parameters)), {
95     H=Sh+Ih+Rh
96     M=Sm+Im
97     dSh <- -(k*p)*((Sh*Im)/(H))+theta*Rh
98     dIh <- (k*p)*((Sh*Im)/(H))-gamma*Ih
99     dRh <- gamma*Ih-theta*Rh
100    dSm <- -(k*q)*((Sm*Ih)/(H))+mu*M-mu*S
101    dIm <- (k*q)*((Sm*Ih)/(H))-mu*Im
102    return(list(c(dSh, dIh, dRh, dSm, dIm)))
103  })
104 }
105 }
106
107 #### Set parameters
108 ## Proportion in each compartment: Susceptible 0.999999, Infected
109 ## 0.000001, Recovered 0
110 init <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
111 ## beta: infection parameter; gamma: recovery parameter
112 parameters <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,
113   theta=1/(365*2))
114 ## Time frame
115 times <- seq(0, 100, by = 1)
116
117 ## Solve using ode (General Solver for Ordinary Differential
118 ## Equations)
119 out <- ode(y = init, times = times, func = sir, parms = parameters
120 )
121 ## change to data frame
122 out <- as.data.frame(out)
123 ## Delete time variable
124 out$time <- NULL
125
126 # Plot deterministic trajectories
127 g<-g+geom_line(data=out, aes_string(x="times",y="Sh"),color='black',
128   , lwd=0.6)+geom_line(data=out, aes_string(x="times",y="Ih"),
129   color='black', lwd=0.6)+geom_line(data=out, aes_string(x="times",
130   ,y="Rh"),color='black', lwd=0.6)
131 g<-g+xlab('Time')
132 g<-g+ylab('Population')
133 print(g)
134
135 h<-h+geom_line(data=out, aes_string(x="times",y="Sm"),color='black',
136   , lwd=1)+geom_line(data=out, aes_string(x="times",y="Im"),
137   color='black', lwd=1)+scale_y_log10()
138 h<-h+xlab('Time')
139 h<-h+ylab('Population (log scale)')
140 h<-h+theme(
141   axis.title.x = element_text(color="black", size=18, face="bold")
142   ,
143   axis.title.y = element_text(color="black", size=18, face="bold")
144 )
145 print(h)

```

Listing 3: malaria_model.r - Main malaria model without adaptations

```

1 #####  

2 # Stochastic malaria model  

3 #####  

4 library(ggplot2)  

5 require(scales)  

6 ## Load deSolve package  

7 library(deSolve)  

8 set.seed(42)  

9 g<-ggplot()  

10 h<-ggplot()  

11 t_init<-0  

12 t_final<-100  

13 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days  

14 x0 <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)  

15 intervals <- seq(from=t_init, to=t_final-dt, by=dt)  

16 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,  

   theta=1/(365*2), K=8000)  

17  

18 # Euler-Maruyama function  

19 em <- function(sir, dt, params) {  

20   H <- sum(sir[1:3]) # Total human pop  

21   M <- sum(sir[4:5]) # total mosquito pop  

22   # Predefine wiener processes (this is a crappy way to do it)  

23   w1<-rnorm(1)*sqrt(dt)  

24   w2<-rnorm(1)*sqrt(dt)  

25   w3<-rnorm(1)*sqrt(dt)  

26   w4<-rnorm(1)*sqrt(dt)  

27   w5<-rnorm(1)*sqrt(dt)  

28   w6<-rnorm(1)*sqrt(dt)  

29   w7<-rnorm(1)*sqrt(dt)  

30  

31   # Make sure no negative values  

32   if(sir['Ih']<0) {  

33     sir['Ih']<-0  

34   }  

35   if(sir['Rh']<0){  

36     sir['Rh']<-0  

37   }  

38  

39   if(sir['Im']<0) {  

40     sir['Im']<-0  

41   }  

42   # Update each compartment (see method in thesis)  

43  

44   # F functions  

45   Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params[  

   'theta']*sir['Rh'])  

46   Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params[  

   'gamma']*sir['Ih'])  

47   Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])  

48   Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+params[  

   'mu']*M-(params['mu']*(sir['Sm'])*(M/params['K'])))  

49   Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-(params[  

   'mu']*(sir['Im'])*(M/params['K'])))
```

```

50
51 # Sigma
52 Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'])
53   ])*w1+sqrt(params['theta']*sir['Rh'])*w2)
53 Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im']))
54   *w1-sqrt(params['gamma']*sir['Ih'])*w3)
54 Rh_sigma<-(sqrt(params['gamma']*sir['Ih']))*w3-sqrt(params['theta']
55   )*sir['Rh'])*w2)
55 Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)
56   )*w4+sqrt(params['mu']*M)*w5-sqrt(params['mu']*(sir['Sm'])*(M/
56   params['K'])))*w6)
56 Im_sigma<-(sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))
57   *w4-sqrt(params['mu']*(sir['Im'])*(M/params['K'])))*w7)
57 # Compute next time step
58 Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
59 Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
60 Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
61 Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
62 Im_next<-sir['Im']+Im_f*dt+Im_sigma
63 # S_next<- sir['S']+(-(params['beta']*sir['S']*sir['I'])/(N)*dt)
63   +sqrt((params['beta']*sir['S']*sir['I'])/(N))*(rnorm(1)*sqrt(dt))
63   ))
64 # I_next<- sir['I']+((params['beta']*sir['S']*sir['I'])/(N)-
64   params['gamma']*sir['I'])*dt+(sqrt((params['beta']*sir['S']*sir
64   ['I'])/(N))-sqrt(params['gamma']*sir['I']))*rnorm(1)*sqrt(dt))
65 # R_next<- N-S_next-I_next # R is easy to recover given constant
65   population
66
67 return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
67   grouping
68 }
69
70 # Run multiple iterations
71 for (iter in seq(1,50)) {
72   set.seed(iter) # Set seed for reproducible results
73   print(iter)
74   df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(), Rh=
74     numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
74     point update
75
76 x=x0 # Set initial conditions
77 # Run single iteration
78 for (delta_t in intervals) {
79   updates<-em(x,dt, params) # Run realisation using EM
80
81   df_temp<-data.frame(time=delta_t,Sh=updates[1], Ih=updates[2],
81     Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in temp.
81     DataFrame
82   df<-rbind(df,df_temp) # Bind to dataframe of all updates
83   x<-updates # Update state variables
84 }
85 # store in output variable then plot
86 out<-df
87 g<-g+geom_line(data=out, aes_string(x="time",y="Sh"),color='red',

```

```

    ,lwd = 0.6, linetype='dotted')+geom_line(data=out, aes_string(x
    ="time",y="Ih"),lwd = 0.6,color='blue',linetype='dotted')+geom_
    line(data=out, aes_string(x="time",y="Rh"),lwd = 0.4,color=
    'green',linetype='dotted')
88 h<-h+geom_line(data=out, aes_string(x="time",y="Im"),lwd = 0.6,
    color='blue', linetype='dotted')+geom_line(data=out, aes_string
    (x="time",y="Sm+Im"),lwd = 0.6,color='pink', linetype='dotted')
89 print(g) # Needed to show plot
90 print(h)
91 }
92
93
94 #####
95 # SIR deterministic model - Adapted from: http://rstudio-pubs-
    static.s3.amazonaws.com/6852_c59c5a2e8ea3456abbeb017185de603e.
    html
96 # FOR BAISC MALARIA MODEL
97 #####
98
99 ## Create an SIR function
100 sir <- function(time, state, parameters) {
101
102   with(as.list(c(state, parameters)), {
103     H=Sh+Ih+Rh
104     M=Sm+Im
105     dSh <- -(k*p)*((Sh*Im)/(H))+theta*Rh
106     dIh <- (k*p)*((Sh*Im)/(H))-gamma*Ih
107     dRh <- gamma*Ih-theta*Rh
108     dSm <- -(k*q)*((Sm*Ih)/(H))+mu*M-mu*Sm*(M/K)
109     dIm <- (k*q)*((Sm*Ih)/(H))-mu*Im*(M/K)
110     return(list(c(dSh, dIh, dRh, dSm, dIm)))
111   })
112 }
113
114 #### Set parameters
115 ## Proportion in each compartment: Susceptible 0.999999, Infected
    0.000001, Recovered 0
116 init      <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
117 ## beta: infection parameter; gamma: recovery parameter
118 parameters <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,
    theta=1/(365*2), K=8000)
119 ## Time frame
120 times      <- seq(0, 100, by = 1)
121
122 ## Solve using ode (General Solver for Ordinary Differential
    Equations)
123 out <- ode(y = init, times = times, func = sir, parms = parameters
    )
124 ## change to data frame
125 out <- as.data.frame(out)
126 ## Delete time variable
127 out$time <- NULL
128 g<-g+geom_line(data=out, aes_string(x="times",y="Sh"),color='black
    ', lwd=0.6)+geom_line(data=out, aes_string(x="times",y="Ih"),

```

```

    color='black', lwd=0.6)+geom_line(data=out, aes_string(x="times
    ",y="Rh"),color='black', lwd=0.6)
129 g<-g+xlab('Time')
130 g<-g+ylab('Population ')
131 print(g)
132
133 #h<-h+geom_line(data=out, aes_string(x="times",y="Im"),color='
    black', lwd=0.6)+geom_line(data=out, aes_string(x="times",y="Sm
    +Im"),color='black', lwd=0.6)+scale_y_log10()
134 h<-h+geom_hline(yintercept=parameters['K'],linetype="dashed")+
    scale_y_log10()
135 h<-h+geom_line(data=out, aes_string(x="times",y="Im"),color='black
    ', lwd=0.6)
136 h<-h+xlab('Time')
137 h<-h+ylab('Population (log scale)')
138 print(h)

```

Listing 4: variable_pop.r - Malaria model with variable population

```

1 ######
2 # Stochastic malaria model
3 #####
4 library(ggplot2)
5
6 ## Load deSolve package
7 library(deSolve)
8 set.seed(42)
9 g<-ggplot()
10 h<-ggplot()
11 t_init<-0
12 t_final<-100
13 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days
14 x0 <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
15 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
16 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu_b=0.033,
    mu_d=0.033, omega=0.004125, theta=1/(365*2))
17
18 # Euler-Maruyama function
19 em <- function(sir, dt, params) {
20   H <- sum(sir[1:3]) # Total human pop
21   M <- sum(sir[4:5]) # total mosquito pop
22   # Predefine wiener processes (this is a crappy way to do it)
23   w1<-rnorm(1)*sqrt(dt)
24   w2<-rnorm(1)*sqrt(dt)
25   w3<-rnorm(1)*sqrt(dt)
26   w4<-rnorm(1)*sqrt(dt)
27   w5<-rnorm(1)*sqrt(dt)
28   w6<-rnorm(1)*sqrt(dt)
29   w7<-rnorm(1)*sqrt(dt)
30   w8<-rnorm(1)*sqrt(dt)
31
32   # Make sure no negative values
33   if(sir['Ih']<0) {
34     sir['Ih']<-0
35   }

```

```

36   if(sir['Rh']<0){
37     sir['Rh']<-0
38   }
39
40   # Update each compartment (see method in thesis)
41
42   # F functions
43   Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params[,
44     'theta']*sir['Rh'])
45   Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params[,
46     'gamma']*sir['Ih'])
47   Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
48   Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+(params[,
49     'mu_b']-params['omega'])*M-params['mu_d']*sir['Sm'])
50   Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-params[,
51     'mu_d']*sir['Im'])
52
53   # Sigma
54   Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'],
55     ])*w1+sqrt(params['theta']*sir['Rh'])*w2)
56   Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im']),
57     *w1-sqrt(params['gamma']*sir['Ih'])*w3)
58   Rh_sigma<-(sqrt(params['gamma']*sir['Ih']))*w3-sqrt(params['theta'],
59     ]*sir['Rh'])*w2)
60   Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H),
61     )*w4+sqrt((params['mu_b'])*M)*w5-sqrt(params['omega']*M)*w8-
62     sqrt(params['mu_d'])*sir['Sm'])*w6)
63   Im_sigma<-(sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H),
64     )*w4-sqrt(params['mu_d'])*sir['Im'])*w7)
65
66   # Compute next time step
67   Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
68   Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
69   Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
70   Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
71   Im_next<-sir['Im']+Im_f*dt+Im_sigma
72   return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
73     grouping
74 }
75
76 # Run multiple iterations
77 for (iter in seq(1,50)) {
78   set.seed(iter) # Set seed for reproducible results
79   print(iter)
80   df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(), Rh=
81     numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
82     point update
83
84   x=x0 # Set initial conditions
85   # Run single iteration
86   for (delta_t in intervals) {
87     updates<-em(x,dt, params) # Run realisation using EM
88
89     df_temp<-data.frame(time=delta_t,Sh=updates[1], Ih=updates[2],
90       Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in temp.

```

```

    DataFrame
  76   df<-rbind(df,df_temp) # Bind to dataframe of all updates
  77   x<-updates # Update state variables
  78 }
  79 # store in output variable then plot
 80 out<-df
 81 g<-g+geom_line(data=out, aes_string(x="time",y="Sh"),color='red',
 82   ,lwd = 0.4, linetype='dotted')+geom_line(data=out, aes_string(x
 83   ="time",y="Ih"),lwd = 0.4,color='blue',linetype='dotted')+geom_
 84   line(data=out, aes_string(x="time",y="Rh"),lwd = 0.4,color='
 85   green',linetype='dotted')
 86 h<-h+geom_line(data=out, aes_string(x="time",y="Sm"),color='red'
 87   ,lwd = 0.4)+geom_line(data=out, aes_string(x="time",y="Im"),lwd
 88   = 0.4,color='blue')
 89 print(g) # Needed to show plot
 90 print(h)
 91 }
 92
 93 #####
 94 # SIR deterministic model - Adapted from: http://rstudio-pubs-
 95   static.s3.amazonaws.com/6852_c59c5a2e8ea3456abbeb017185de603e.
 96   html
 97 # FOR BAISC MALARIA MODEL
 98 #####
 99
100 ## Create an SIR function
101 sir <- function(time, state, parameters) {
102
103   with(as.list(c(state, parameters)), {
104     H=Sh+Ih+Rh
105     M=Sm+Im
106     dSh <- -(k*p)*((Sh*Im)/(H))+theta*Rh
107     dIh <- (k*p)*((Sh*Im)/(H))-gamma*Ih
108     dRh <- gamma*Ih-theta*Rh
109     dSm <- -(k*q)*((Sm*Ih)/(H))+(mu_b-omega)*M-mu_d*Sm
110     dIm <- (k*q)*((Sm*Ih)/(H))-mu_d*Im
111     return(list(c(dSh, dIh, dRh, dSm, dIm)))
112   })
113 }
114
115 #### Set parameters
116 ## Proportion in each compartment: Susceptible 0.999999, Infected
117   0.000001, Recovered 0
118 init      <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
119 ## beta: infection parameter; gamma: recovery parameter
120 parameters <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu_b
121   =0.033, mu_d=0.033, omega=0.004125, theta=1/(365*2))
122 ## Time frame
123 times      <- seq(0, 100, by = 1)
124
125 ## Solve using ode (General Solver for Ordinary Differential
126   Equations)
127 out <- ode(y = init, times = times, func = sir, parms = parameters

```

```

    )
118 ## change to data frame
119 out <- as.data.frame(out)
120 ## Delete time variable
121 out$time <- NULL
122 g<-g+geom_line(data=out, aes_string(x="times",y="Sh"),color='black',
123   , lwd=0.6)+geom_line(data=out, aes_string(x="times",y="Ih"),
124   color='black', lwd=0.6)+geom_line(data=out, aes_string(x="times",
125   "y="Rh"),color='black', lwd=0.6)
126 g<-g+xlab('Time')
127 g<-g+ylab('Population')
128 print(g)
129
130 h<-h+geom_line(data=out, aes_string(x="times",y="Sm"),color='black',
131   , lwd=1)+geom_line(data=out, aes_string(x="times",y="Im"),
132   color='black', lwd=1)+scale_y_log10()
133 h<-h+xlab('Time')
134 h<-h+ylab('Population (log scale)')
135 h<-h+theme(
136   axis.title.x = element_text(color="black", size=14, face="bold")
137   ,
138   axis.title.y = element_text(color="black", size=14, face="bold"))
139 )
140 print(h)

```

Listing 5: wolbachia.r - malaria model with Wolbachia introduction

```

1 #####
2 # Stochastic malaria model
3 # Basic
4 #####
5 library(ggplot2)
6
7 ## Load deSolve package
8 library(deSolve)
9 set.seed(42)
10 g<-ggplot()
11 h<-ggplot()
12 t_init<-0
13 t_final<-100
14 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days
15 x0 <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
16 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
17 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,
18   theta=1/(365*2))
19
20 # Euler-Maruyama function
21 em <- function(sir, dt, params) {
22   H <- sum(sir[1:3]) # Total human pop
23   M <- sum(sir[4:5]) # total mosquito pop
24   # Predefine wiener processes (this is a crappy way to do it)
25   w1<-rnorm(1)*sqrt(dt)
26   w2<-rnorm(1)*sqrt(dt)
27   w3<-rnorm(1)*sqrt(dt)
28   w4<-rnorm(1)*sqrt(dt)

```

```

28 w5<-rnorm(1)*sqrt(dt)
29 w6<-rnorm(1)*sqrt(dt)
30 w7<-rnorm(1)*sqrt(dt)
31
32 # Make sure no negative values
33 if(sir['Ih']<0) {
34   sir['Ih']<-0
35 }
36 if(sir['Rh']<0){
37   sir['Rh']<-0
38 }
39
40 # Update each compartment (see method in thesis)
41
42 # F functions
43 Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params[
44   'theta']*sir['Rh'])
45 Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params['
46   gamma']*sir['Ih'])
47 Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
48 Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+params[
49   'mu']*M-params['mu']*sir['Sm'])
50 Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-params['
51   mu']*sir['Im'])
52
53 # Sigma
54 Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'
55   ])*w1+sqrt(params['theta']*sir['Rh'])*w2)
56 Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'
57   ])*w1-sqrt(params['gamma']*sir['Ih'])*w3)
58 Rh_sigma<-(sqrt(params['gamma']*sir['Ih']))*w3-sqrt(params['
59   theta']*sir['Rh'])*w2)
60 Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H
61   ))*w4+sqrt(params['mu']*M)*w5-sqrt(params['mu']*sir['Sm'])*w6)
62 Im_sigma<-(sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H
63   ))*w4-sqrt(params['mu']*sir['Im'])*w7)
64
65 # Compute next time step
66 Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
67 Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
68 Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
69 Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
70 Im_next<-sir['Im']+Im_f*dt+Im_sigma
71
72 return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
73 grouping
74 }
75 param_change<-c(0.42,0.63,0.315)
76 param_change_linetype<-c("blue","red",'green')
77 # Run iterations for changing parameters
78 for (val in seq(1,3)) {
79   params['k']=param_change[val]
80   # Run multiple iterations
81   for (iter in seq(1,3)) {
82     print(iter)

```

```

72   df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(), Rh=
73     numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
74     point update
75
76   x=x0 # Set initial conditions
77   # Run single iteration
78   for (delta_t in intervals) {
79     updates<-em(x,dt, params) # Run realisation using EM
80
81     df_temp<-data.frame(time=delta_t,Sh=updates[1],Ih=updates
82     [2],Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in
83     temp. DataFrame
84     df<-rbind(df,df_temp) # Bind to dataframe of all updates
85     x<-updates # Update state variables
86   }
87
88   # store in output variable then plot
89   out<-df
90   g<-g+geom_line(data=out, aes_string(x="time",y="Ih"),lwd =
91     0.5,color=param_change_linetype[val])
92   g<-g+xlab('Time')
93   g<-g+ylab('Infected Human Population')
94   print(g) # Needed to show plot
95 }
96 #####
97 # Stochastic malaria model
98 # Variable population tbd.
99 #####
100 library(ggplot2)
101
102 ## Load deSolve package
103 library(deSolve)
104 require(scales)
105 set.seed(42)
106 g<-ggplot()
107 h<-ggplot()
108 t_init<-0
109 t_final<-100
110 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days
111 x0 <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
112 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
113 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,
114   theta=1/(365*2), K=8000)
115
116 # Euler-Maruyama function
117 em <- function(sir, dt, params) {
118   H <- sum(sir[1:3]) # Total human pop
119   M <- sum(sir[4:5]) # total mosquito pop
120   # Predefine wiener processes (this is a crappy way to do it)
121   w1<-rnorm(1)*sqrt(dt)
122   w2<-rnorm(1)*sqrt(dt)
123   w3<-rnorm(1)*sqrt(dt)
124   w4<-rnorm(1)*sqrt(dt)

```

```

120   w5<-rnorm(1)*sqrt(dt)
121   w6<-rnorm(1)*sqrt(dt)
122   w7<-rnorm(1)*sqrt(dt)
123
124   # Make sure no negative values
125   if(sir['Ih']<0) {
126     sir['Ih']<-0
127   }
128   if(sir['Rh']<0){
129     sir['Rh']<-0
130   }
131
132   if(sir['Im']<0) {
133     sir['Im']<-0
134   }
135
136   # Update each compartment (see method in thesis)
137
138   # F functions
139   Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params['theta']*sir['Rh'])
140   Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params['gamma']*sir['Ih'])
141   Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
142   Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+params['mu']*M-(params['mu']*(sir['Sm']))*(M/params['K'])))
143   Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-(params['mu']*(sir['Im'])*(M/params['K'])))
144
145   # Sigma
146   Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'])*w1+sqrt(params['theta']*sir['Rh'])*w2)
147   Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'])*w1-sqrt(params['gamma']*sir['Ih'])*w3)
148   Rh_sigma<-(sqrt(params['gamma']*sir['Ih'])*w3-sqrt(params['theta']*sir['Rh'])*w2)
149   Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))*w4+sqrt(params['mu']*M)*w5-sqrt(params['mu']*(sir['Sm'])*(M/params['K'])))*w6)
150   Im_sigma<-(sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))*w4-sqrt(params['mu']*(sir['Im'])*(M/params['K'])))*w7)
151
152   # Compute next time step
153   Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
154   Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
155   Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
156   Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
157   Im_next<-sir['Im']+Im_f*dt+Im_sigma
158   return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return grouping
159
160   # Run multiple iterations
161   param_change<-c(0.074,0.111,0.0555)
162   param_change_linetype<-c("blue","red",'green')
163   # Run iterations for changing parameters

```

```

163 for (val in seq(1,3)) {
164   params[ 'gamma' ]=param_change[val]
165   # Run multiple iterations
166   for (iter in seq(1,3)) {
167     print(iter)
168     df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(), Rh
169 =numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
170     point update
171
172     x=x0 # Set initial conditions
173     # Run single iteration
174     for (delta_t in intervals) {
175       updates<-em(x,dt, params) # Run realisation using EM
176
177       df_temp<-data.frame(time=delta_t,Sh=updates[1], Ih=updates
178 [2],Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in
179       temp. DataFrame
180       df<-rbind(df,df_temp) # Bind to dataframe of all updates
181       x<-updates # Update state variables
182     }
183
184     # store in output variable then plot
185     out<-df
186     g<-g+geom_line(data=out, aes_string(x="time",y="Ih"),lwd =
187 0.5,color=param_change_linetype[val])
188     g<-g+xlab('Time')
189     g<-g+ylab('Infected Human Population')
190     print(g) # Needed to show plot
191   }
192 }
193 # #####
194 # Stochastic malaria model
195 # Wolbachia population tbd.
196 ######
197 library(ggplot2)
198 # Load deSolve package
199 library(deSolve)
200 require(scales)
201 set.seed(42)
202 g<-ggplot()
203 h<-ggplot()
204 t_init<-0
205 t_final<-100
206 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days
207 x0 <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
208 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
209 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu_b=0.033,
210   mu_d=0.033, omega=0.004125, theta=1/(365*2))
211
212 # Euler-Maruyama function
213 em <- function(sir, dt, params) {
214   H <- sum(sir[1:3]) # Total human pop
215   M <- sum(sir[4:5]) # total mosquito pop
216   # Predefine wiener processes (this is a crappy way to do it)

```

```

211 w1<-rnorm(1)*sqrt(dt)
212 w2<-rnorm(1)*sqrt(dt)
213 w3<-rnorm(1)*sqrt(dt)
214 w4<-rnorm(1)*sqrt(dt)
215 w5<-rnorm(1)*sqrt(dt)
216 w6<-rnorm(1)*sqrt(dt)
217 w7<-rnorm(1)*sqrt(dt)
218 w8<-rnorm(1)*sqrt(dt)

219
220 # Make sure no negative values
221 if(sir['Ih']<0) {
222   sir['Ih']<-0
223 }
224 if(sir['Rh']<0){
225   sir['Rh']<-0
226 }
227
228 if(sir['Im']<0) {
229   sir['Im']<-0
230 }
231 # Update each compartment (see method in thesis)

232
233 # F functions
234 Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params[,
235   'theta']*sir['Rh'])
236 Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params[,
237   'gamma']*sir['Ih'])
238 Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
239 Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+(params[
240   'mu_b']-params['omega'])*M-params['mu_d']*sir['Sm']))
241 Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-params[,
242   'mu_d']*sir['Im'])

243
244 # Sigma
245 Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'],
246   ])*w1+sqrt(params['theta']*sir['Rh']))*w2)
247 Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im']),
248   *w1-sqrt(params['gamma']*sir['Ih']))*w3)
249 Rh_sigma<-(sqrt(params['gamma']*sir['Ih']))*w3-sqrt(params['theta'],
250   *sir['Rh'])*w2)
251 Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H),
252   )*w4+sqrt((params['mu_b'])*M)*w5-sqrt(params['omega'])*M)*w8-
253   sqrt(params['mu_d'])*sir['Sm'])*w6)
254 Im_sigma<-(sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H),
255   *w4-sqrt(params['mu_d'])*sir['Im']])*w7)
256
257 # Compute next time step
258 Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
259 Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
260 Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
261 Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
262 Im_next<-sir['Im']+Im_f*dt+Im_sigma
263 return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
264   grouping
265 }

```

```

254 # Run multiple iterations
255 param_change<-c(0.42,0.63,0.315)
256 param_change_linetype<-c("blue","red",'green')
257 # Run iterations for changing parameters
258 for (val in seq(1,3)) {
259   params['k']=param_change[val]
260   # Run multiple iterations
261   for (iter in seq(1,3)) {
262     print(iter)
263     df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(), Rh=
264       numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
265       point update
266
267     x=x0 # Set initial conditions
268     # Run single iteration
269     for (delta_t in intervals) {
270       updates<-em(x,dt, params) # Run realisation using EM
271
272       df_temp<-data.frame(time=delta_t,Sh=updates[1],Ih=updates
273 [2],Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in
274       temp. DataFrame
275       df<-rbind(df,df_temp) # Bind to dataframe of all updates
276       x<-updates # Update state variables
277     }
278
279     # store in output variable then plot
280     out<-df
281     g<-g+geom_line(data=out, aes_string(x="time",y="Ih"),lwd =
282       0.5,color=param_change_linetype[val])
283     g<-g+xlab('Time')
284     g<-g+ylab('Infected Human Population')
285     print(g) # Needed to show plot
286   }
287 }

```

Listing 6: sensitivityanalysis.r - Sensitivity analysis on R0 for all models

```

1 # #####
2 # # Stochastic malaria model
3 # # Wolbachia population tbd.
4 # #####
5 library(ggplot2)
6 ## Load deSolve package
7 library(deSolve)
8 require(scales)
9 set.seed(42)
10 g<-ggplot()
11 h<-ggplot()
12 t_init<-0
13 t_final<-100
14 dt<-(t_final-t_init)/(24*100) # Update every hour for 100 days
15 x0 <- c(Sh = 500, Ih = 0, Rh=0, Sm=3900, Im=100)
16 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
17 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu_b=0.033,

```

```

    mu_d=0.033, omega=0.004125, theta=1/(365*2))

18
19 # Euler-Maruyama function
20 em <- function(sir, dt, params) {
21   H <- sum(sir[1:3]) # Total human pop
22   M <- sum(sir[4:5]) # total mosquito pop
23   # Predefine wiener processes (this is a crappy way to do it)
24   w1<-rnorm(1)*sqrt(dt)
25   w2<-rnorm(1)*sqrt(dt)
26   w3<-rnorm(1)*sqrt(dt)
27   w4<-rnorm(1)*sqrt(dt)
28   w5<-rnorm(1)*sqrt(dt)
29   w6<-rnorm(1)*sqrt(dt)
30   w7<-rnorm(1)*sqrt(dt)
31   w8<-rnorm(1)*sqrt(dt)

32
33   # Make sure no negative values
34   if(sir['Ih']<0) {
35     sir['Ih']<-0
36   }
37   if(sir['Rh']<0){
38     sir['Rh']<-0
39   }
40
41   if(sir['Im']<0) {
42     sir['Im']<-0
43   }
44   # Update each compartment (see method in thesis)
45
46   # F functions
47   Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params['theta']*sir['Rh'])
48   Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params['gamma']*sir['Ih'])
49   Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
50   Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+(params['mu_b']-params['omega'])*M-params['mu_d']*sir['Sm'])
51   Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-params['mu_d']*sir['Im'])

52
53   # Sigma
54   Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'])*w1+sqrt(params['theta']*sir['Rh'])*w2)
55   Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'])*w1-sqrt(params['gamma']*sir['Ih'])*w3)
56   Rh_sigma<-(sqrt(params['gamma']*sir['Ih'])*w3-sqrt(params['theta']*sir['Rh'])*w2)
57   Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))*w4+sqrt((params['mu_b'])*M)*w5-sqrt(params['omega']*M)*w8-sqrt(params['mu_d']*sir['Sm'])*w6)
58   Im_sigma<-(sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))*w4-sqrt(params['mu_d']*sir['Im'])*w7)
59   # Compute next time step
60   Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma

```

```

61   Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
62   Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
63   Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
64   Im_next<-sir['Im']+Im_f*dt+Im_sigma
65   return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
       grouping
66 }
67
68 # Run multiple iterations
69 param_change<-c(0,33/8000,33/4000)
70 param_change_linetype<-c('green', 'blue', 'red')
71 # Run iterations for changing parameters
72 for (val in seq(1,length(param_change))) {
73   params['omega']=param_change[val]
74   # Run multiple iterations
75   for (iter in seq(1,3)) {
76     print(iter)
77     df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(),
78                      Rh=numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
79     point update
80
81     x=x0 # Set initial conditions
82     # Run single iteration
83     for (delta_t in intervals) {
84       updates<-em(x,dt, params) # Run realisation using EM
85
86       df_temp<-data.frame(time=delta_t, Sh=updates[1], Ih=updates
87       [2], Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in
88       temp. DataFrame
89       df<-rbind(df,df_temp) # Bind to dataframe of all updates
90       x<-updates # Update state variables
91     }
92
93     # store in output variable then plot
94     out<-df
95     g<-g+geom_line(data=out, aes_string(x="time",y="Sm"),lwd =
96       0.5,color=param_change_linetype[val])
97     g<-g+xlab('Time')
98     g<-g+ylab('Susceptible Mosquito population')
99     print(g) # Needed to show plot
100   }
101 }

```

Listing 7: sensitivityanalysis_wolbachia.r - Sensitivity analysis for the rate of wolbachia in mosquito population

```

1 library(ggplot2)
2 library(dplyr)
3 # Plot case study cases
4 cases <- c(24, 49, 64, 93, 117, 154, 195, 242, 272, 299, 317, 332)
5 month <- c("1993-01", "1993-02", "1993-03", "1993-04", "1993-05",
6      "1993-06", "1993-07", "1993-08", "1993-09", "1993-10", "1993-11",
7      "1993-12")
8 case_study <- data.frame(cases, month)
9 g <- ggplot()

```

```

8 # Stochastic simulation
9 set.seed(42)
10 g<-ggplot()
11 h<-ggplot()
12 t_init<-0
13 t_final<-365
14 dt<-(t_final-t_init)/(24*365) # Update every hour for 365 days
15 x0 <- c(Sh = 90000, Ih = 0, Rh=0, Sm=90000*8, Im=100)
16 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
17 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,
18   theta=1/(365*2))
19
20 # Euler-Maruyama function
21 em <- function(sir, dt, params) {
22   H <- sum(sir[1:3]) # Total human pop
23   M <- sum(sir[4:5]) # total mosquito pop
24   # Predefine wiener processes (this is a crappy way to do it)
25   w1<-rnorm(1)*sqrt(dt)
26   w2<-rnorm(1)*sqrt(dt)
27   w3<-rnorm(1)*sqrt(dt)
28   w4<-rnorm(1)*sqrt(dt)
29   w5<-rnorm(1)*sqrt(dt)
30   w6<-rnorm(1)*sqrt(dt)
31   w7<-rnorm(1)*sqrt(dt)
32
33   # Make sure no negative values
34   if(sir['Ih']<0) {
35     sir['Ih']<-0
36   }
37   if(sir['Rh']<0){
38     sir['Rh']<-0
39   }
40
41   # Update each compartment (see method in thesis)
42
43   # F functions
44   Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params[',
45     'theta']*sir['Rh'])
46   Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params[',
47     'gamma']*sir['Ih'])
48   Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
49   Sm_f<-(-params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+params[',
50     'mu']*M-params['mu']*sir['Sm'])
51   Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-params[',
52     'mu']*sir['Im'])
53
54   # Sigma
55   Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'],
56     ])*w1+sqrt(params['theta']*sir['Rh'])*w2)
57   Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'],
58     ])*w1-sqrt(params['gamma']*sir['Ih'])*w3)
59   Rh_sigma<-(sqrt(params['gamma']*sir['Ih'])*w3-sqrt(params['theta'],
60     ])*sir['Rh'])*w2)
61   Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))

```

```

      ) *w4+sqrt(params['mu']*M)*w5-sqrt(params['mu']*sir['Sm'])*w6)
54  Im_sigma<- (sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H))
55   *w4-sqrt(params['mu']*sir['Im'])*w7)
# Compute next time step
56  Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
57  Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
58  Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
59  Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
60  Im_next<-sir['Im']+Im_f*dt+Im_sigma
61  # S_next<- sir['S']+(-(params['beta']*sir['S']*sir['I'])/(N)*dt)
62   +sqrt((params['beta']*sir['S']*sir['I'])/(N))*(rnorm(1)*sqrt(dt))
63   )
# I_next<- sir['I']+((params['beta']*sir['S']*sir['I'])/(N)-
64   params['gamma']*sir['I'])*dt+(sqrt((params['beta']*sir['S']*sir
65   ['I'])/(N))-sqrt(params['gamma']*sir['I']))*rnorm(1)*sqrt(dt)
66  # R_next<- N-S_next-I_next # R is easy to recover given constant
67   population
68
69  return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
70   grouping
71 }
72
73 # Run multiple iterations
74 for (iter in seq(1,1)) {
75   set.seed(iter) # Set seed for reproducible results
76   print(iter)
77   df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(),
78     numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
79   point update
80
81   x=x0 # Set initial conditions
82   # Run single iteration
83   for (delta_t in intervals) {
84     updates<-em(x,dt, params) # Run realisation using EM
85
86     df_temp<-data.frame(time=delta_t, Sh=updates[1], Ih=updates[2],
87     Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in temp.
88     DataFrame
89     df<-rbind(df,df_temp) # Bind to dataframe of all updates
90     x<-updates # Update state variables
91   }
92   # store in output variable then plot
93   out<-df
94   culm_cases <- vector(mode = "list", length = 12)
95   i <- 1
96   culm_cases[1] <- 0
97   for (val in c(31*24,59*24,90*24,120*24,151*24,181*24,212*24,243*
98     24,273*24,304*24,334*24,365*24)) {
99     culm_cases[i+1] <- culm_cases[[i]]+out$Ih[val]
  i<- i+1
}
simulation_res <- data.frame(Reduce(rbind, culm_cases[2:13]),
  month)
g<- g+ geom_line(data=simulation_res, aes(x=month, y=Reduce.

```

```

  rbind..culm_cases.2.13.., group=1), color='blue')+geom_point(
    data=case_study, aes(x=month, y=cases, group=1), shape=4, color=
      'red')+theme(axis.text.x = element_text(angle = 45, vjust =
        0.5, hjust=1))
94 g<-g+xlab('Month')
95 g<-g+ylab('Cases')
96 g<-g+theme(
97   axis.title.x = element_text(color="black", size=16, face="bold"
98   ),
99   axis.title.y = element_text(color="black", size=16, face="bold"
100 )
101 print(g)
102 }
```

Listing 8: modelvalidation.r - Model validation plotting

```

1 library(ggplot2)
2 library(dplyr)
3 # Plot case study cases
4 cases <- c(24, 49, 64, 93, 117, 154, 195, 242, 272, 299, 317, 332)
5 month <- c("1993-01", "1993-02", "1993-03", "1993-04", "1993-05",
      "1993-06", "1993-07", "1993-08", "1993-09", "1993-10", "1993-11",
      ", "1993-12")
6 case_study <- data.frame(cases, month)
7 g <- ggplot()
8 total_errors<-data.frame(month=month)
9 # Stochastic simulation
10 set.seed(42)
11 t_init<-0
12 t_final<-365
13 dt<-(t_final-t_init)/(24*365) # Update every hour for 365 days
14 x0 <- c(Sh = 90000, Ih = 0, Rh=0, Sm=485820, Im=100)
15 intervals <- seq(from=t_init, to=t_final-dt, by=dt)
16 params <- c(k=0.42, p=0.07, q=0.0375, gamma = 0.074, mu=0.033,
     theta=1/(365*2))
17
18 # Euler-Maruyama function
19 em <- function(sir, dt, params) {
20   H <- sum(sir[1:3]) # Total human pop
21   M <- sum(sir[4:5]) # total mosquito pop
22   # Predefine wiener processes (this is a crappy way to do it)
23   w1<-rnorm(1)*sqrt(dt)
24   w2<-rnorm(1)*sqrt(dt)
25   w3<-rnorm(1)*sqrt(dt)
26   w4<-rnorm(1)*sqrt(dt)
27   w5<-rnorm(1)*sqrt(dt)
28   w6<-rnorm(1)*sqrt(dt)
29   w7<-rnorm(1)*sqrt(dt)
30
31   # Make sure no negative values
32   if(sir['Ih']<0) {
33     sir['Ih']<-0
34   }
35   if(sir['Rh']<0){
```

```

36     sir['Rh'] <- 0
37 }
38
39 # Update each compartment (see method in thesis)
40
41 # F functions
42 Sh_f<-(-params['p']*params['k']*(sir['Sh']/H)*sir['Im']+params[,
43   'theta']*sir['Rh'])
44 Ih_f<-(params['p']*params['k']*(sir['Sh']/H)*sir['Im']-params[,
45   'gamma']*sir['Ih'])
46 Rh_f<-(params['gamma']*sir['Ih']-params['theta']*sir['Rh'])
47 Sm_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)+params[,
48   'mu']*M-params['mu']*sir['Sm'])
49 Im_f<-(params['q']*params['k']*sir['Sm']*(sir['Ih']/H)-params[,
50   'mu']*sir['Im'])
51
52 # Sigma
53 Sh_sigma<-(-sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im'],
54   ])*w1+sqrt(params['theta']*sir['Rh'])*w2)
55 Ih_sigma<-(sqrt(params['p']*params['k']*(sir['Sh']/H)*sir['Im']),
56   *w1-sqrt(params['gamma']*sir['Ih'])*w3)
57 Rh_sigma<-(sqrt(params['gamma']*sir['Ih'])*w3-sqrt(params['theta'],
58   ])*sir['Rh'])*w2)
59 Sm_sigma<-(-sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H),
60   )*w4+sqrt(params['mu']*M)*w5-sqrt(params['mu']*sir['Sm'])*w6)
61 Im_sigma<-(sqrt(params['q']*params['k']*sir['Sm']*(sir['Ih']/H),
62   )*w4-sqrt(params['mu']*sir['Im']))*w7)
63
64 # Compute next time step
65 Sh_next<-sir['Sh']+Sh_f*dt+Sh_sigma
66 Ih_next<-sir['Ih']+Ih_f*dt+Ih_sigma
67 Rh_next<-sir['Rh']+Rh_f*dt+Rh_sigma
68 Sm_next<-sir['Sm']+Sm_f*dt+Sm_sigma
69 Im_next<-sir['Im']+Im_f*dt+Im_sigma
70 # S_next<- sir['S']+(-(params['beta']*sir['S']*sir['I'])/(N)*dt)
71 # +sqrt((params['beta']*sir['S']*sir['I'])/(N))*(rnorm(1)*sqrt(dt))
72 # I_next<- sir['I']+((params['beta']*sir['S']*sir['I'])/(N)-
73 #   params['gamma']*sir['I'])*dt+(sqrt((params['beta']*sir['S']*sir
74 #   ['I'])/(N))-sqrt(params['gamma']*sir['I']))*rnorm(1)*sqrt(dt)
75 # R_next<- N-S_next-I_next # R is easy to recover given constant
76 # population
77
78 return(c(Sh_next,Ih_next,Rh_next,Sm_next,Im_next)) # Return
79   grouping
80 }
81
82
83 # Run multiple iterations
84 for (iter in seq(1,25)) {
85   set.seed(iter) # Set seed for reproducible results
86   print("Iteration: ")
87   print(iter)
88   df<-data.frame(time=numeric(), Sh=numeric(), Ih=numeric(), Rh=
89     numeric(), Sm=numeric(), Im=numeric()) # Used to store each time
90   point update

```

```

73
74   x=x0 # Set initial conditions
75   # Run single iteration
76   for (delta_t in intervals) {
77     updates<-em(x,dt, params) # Run realisation using EM
78
79     df_temp<-data.frame(time=delta_t,Sh=updates[1],Ih=updates[2],
80     Rh=updates[3], Sm=updates[4], Im=updates[5]) # store in temp.
81     DataFrame
82     df<-rbind(df,df_temp) # Bind to dataframe of all updates
83     x<-updates # Update state variables
84   }
85   # store in output variable then plot
86   out<-df
87   culm_cases <- vector(mode = "list", length = 12)
88   i <- 1
89   culm_cases[1] <- 0
90   for (val in c(31*24,59*24,90*24,120*24,151*24,181*24,212*24,243*
91     24,273*24,304*24,334*24,365*24)) {
92     culm_cases[i+1] <- culm_cases[[i]]+out$Ih[val]
93     i<- i+1
94   }
95   simulation_res <- data.frame(col=Reduce(rbind, culm_cases[2:13]))
96
97   simulation_error<- abs(simulation_res$col-case_study$cases)
98   total_errors <- cbind(total_errors,simulation_error)
99 }
100
101 boxplot_error<-data.frame(t(total_errors))
102 names(boxplot_error)<- boxplot_error[1,]
103 boxplot_error<-boxplot_error[-1,]
104 boxplot_error<-lapply(boxplot_error,as.numeric)
105 boxplot(boxplot_error)
106 title(xlab="Months", ylab="Absolute Error")

```

Listing 9: modelvalidation_error.r - Model validation boxplot error after 50 realisations